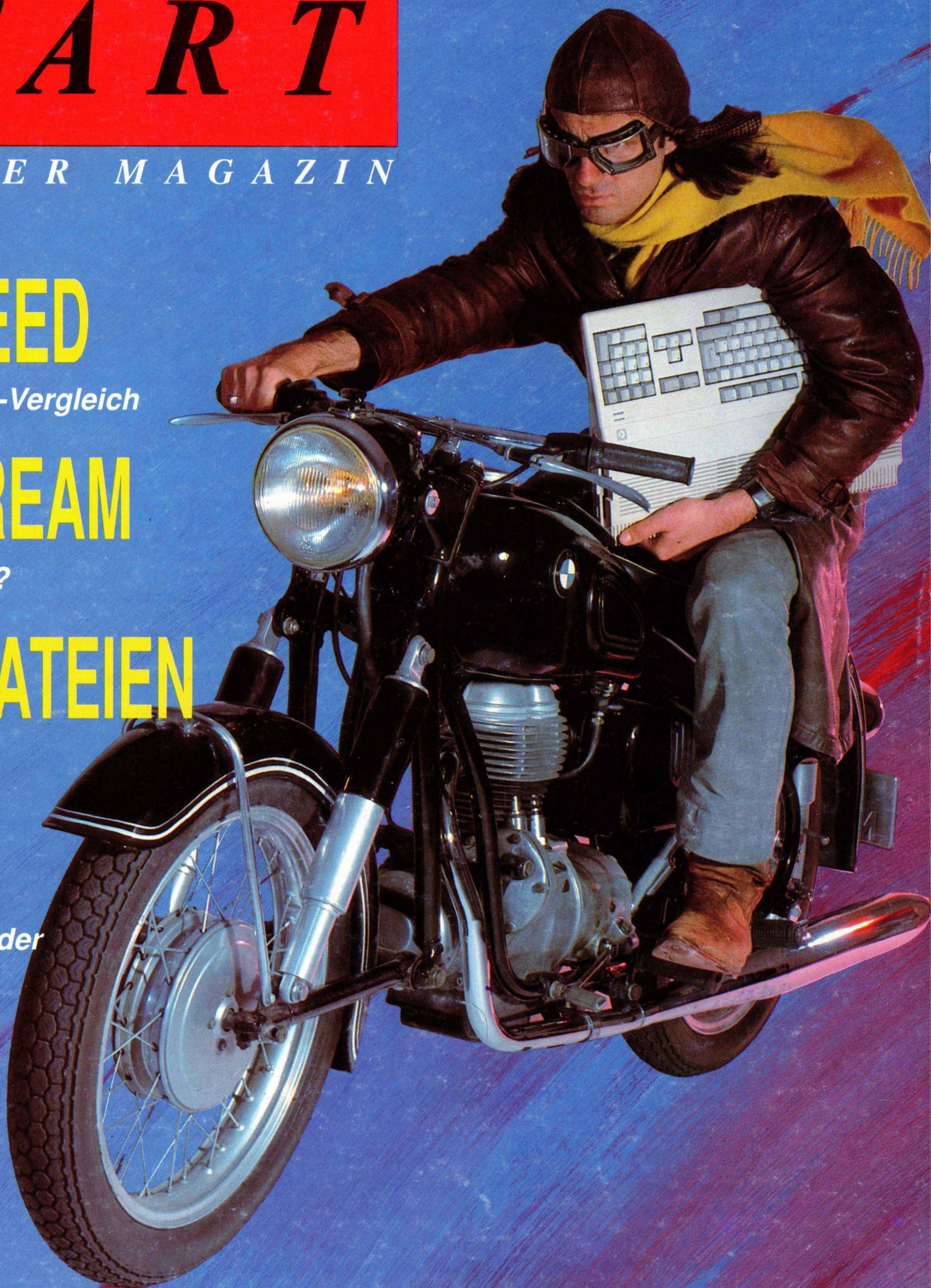
DM 7,- Sfr 7.- Ös 56.- Juni ´89 Nr. 6

AMIGA

# KICK START

COMPUTER MAGAZIN



## FULL SPEED
Großer Assembler-Vergleich

## PAGESTREAM
Neuer DTP-Traum?

## BATCH-DATEIEN
Die großen Helfer

## LISTINGS
Optimierter IFF-Lader
Guru-Musik
Virusfalle
Grafikeffekte

BTX auf AMIGA • WShell • Tips & Tricks
Spiele • Lösungen • Drucker • Neuheiten

# AMIGA-NEWS 6/89

## Ein Informationsservice von Ihren Distributoren DTM Deutschland und MICROTRON Schweiz

## Unsere Bestseller

Z.Zt. führen wir Produkte der Firmen AEGIS, AMERICAN COVERS, ANAKIN RESEARCH, BYTE BY BYTE, CENTRAL COAST SOFTWARE, DTM, ELECTRONIC ARTS, IDEAS CREATED, LATTICE, METACOMCO, MICRO WAY, MINDWARE INTERNATIONAL, PRECISION SOFTWARE, GREAT VALLEY PRODUCTS, VIZA SOFTWARE.
Hier eine Auswahl der derzeitigen Bestseller:

| | |
|---|---|
| AEGIS alle Produkte | lieferbar |
| Animate 3D | 298,- DM |
| Calligrapher | 229,- DM |
| Digi View Gold deutsch | 398,- DM |
| Digi Paint deutsch | 148,- DM |
| DISCovery deutsch | 198,- DM |
| Disk to Disk | 109,- DM |
| DOS to DOS deutsch | 128,- DM |
| EASYL Zeichentabletts | |
| Easyl Amiga 500 | 698,- DM |
| Easyl Amiga 1000 | 798,- DM |
| Easyl Amiga 2000 | 898,- DM |
| Funktion deutsch | 98,- DM |
| Jitter Rid Filterscheibe | 29,95 DM |
| Pageflipper deutsch | 77,- DM |
| Pageflipper F/X plus PAL | 398,- DM |
| Quarterback deutsch | 128,- DM |
| Sculpt/Animate 4D | 998,- DM |

**HANDBÜCHER einzeln:**

| | |
|---|---|
| AEGIS Audiomaster | 29,95 DM |
| AEGIS Sonix | 49,95 DM |
| AEGIS Videotitler | 39,95 DM |
| Balance of Power | 29,95 DM |
| Calligrapher | 29,95 DM |
| Comicsetter | 29,95 DM |
| DOS to DOS | 19,95 DM |
| Flugsimulator II | 29,95 DM |
| Jet | 29,95 DM |
| Kampfgruppe | 29,95 DM |

## Neue AEGIS Produkte

### ANIMagic

ein leistungsstarkes Werkzeug für jede erdenkliche Art von Bildmanipulation. Lassen Sie die Bilder in Farbexplosionen verschwinden, neue Bilder aus Schatten entstehen oder wandeln Sie durch eine transparent werdende Grafik. Eine Vielzahl an digitalen Effekten, kombiniert mit Animation geben neue, ungeahnte Möglichkeiten der Bildgestaltung. ANIMagic kombiniert Hunderte von Effekten zusammen mit dem SPARTA/ANIM Format, das zusammen mit Aegis von vielen populären Programmen unterstützt wird, z.B. The Director, Deluxe Paint II, Photon Cell Animator und Hash Animation Series.
ANIMagic benötigt mindestens 2 MByte Hauptspeicher und kostet 228,- DM

## SoundTrax I & II

Eine neue Qualität im Soundbereich erleben Sie mit den neuen SoundTrax Disketten von AEGIS. Professionelle Samples von ausgesuchten Instrumenten bieten eine klangreine Basis für alle zukünftigen Sonix Anwendungen.

SoundTrax Disketten sind nur mit Aegis SONIX lauffähig und kosten je Diskette 49,- DM

## Die erste SCSI-Festplatte für den Amiga 500

- **Voll autobootfähig mit Kickstart 1.3, bootet direkt von der Fast-File Partition**
- **Schnelles 3,5 Zoll SCSI Drive im Lieferumfang**
- **Ausbaufähig auf 2 MB Fast-RAM durch Einstecken von Megabitchips**
- **Externes Netzteil, dadurch kein Wärmestau und Störungen im Plattengehäuse**
- **Durchgeschleifter SCSI-Bus für bis zu 6 weiteren Geräten, z.B. optische Platte etc.**
- **Deutsche Installationssoftware mit deutscher Anleitung**
- **Sensationeller Preis - 20 MByte 1595,- DM größere Kapazitäten auf Anfrage**




### Weitere GVP Produkte für AMIGA 2000

IMPACT SCSI-Hardcards A-2000:

| | |
|---|---|
| IMPACT 30 MB 28msec. | 1698,- |
| IMPACT 40 MB 28msec. | 1898,- |
| IMPACT 40 MB 11msec. | 2295,- |
| IMPACT 80 MB 11msec. | 3195,- |

SCSI Controller A-2000 einzeln:

| | |
|---|---|
| IMPACT SCSI 2/0 ohne RAM | 698,- |
| IMPACT SCSI 2/2 mit 2MB | 1898,- |
| Autoboot-Kit | 50,- |

Weitere Karten A-2000:

| | |
|---|---|
| Turbo-Board 68030/25Mhz incl. HD-Controller | 2498,- |
| Memory 4MB/32bit | 3998,- |

## Der schnellste Amiga der Welt

läuft derzeit in unserem Schweizer Testlabor unter härtesten Testbedingungen. Wie versprochen, erhalten Sie heute einige Daten und Informationen. Das Turboboard wird voraussichtlich ab Mitte Juni 1989 ausgeliefert.



- **Hi-Performance Prozessor Motorola 68030 mit 25 Megahertz Taktfrequenz**
- **Asynchrones Timing erlaubt variable Taktfrequenz unabhängig vom Amiga**
- **Aufrüstbar mit 8 Megabyte, 32 bit, autokonfiguriendem D-RAM Speicher**
- **Unterstützt den 68030 Burst-Mode, dadurch 0 waitstate RAM Zugriff bei 25 MHz!**
- **Umschaltbar in 68000er Betrieb (Software- und Hardwaremäßig)**
- **Eingebaute Memory-Management Unit (MMU) für höhere Anwendungen, z.B. UNIX**
- **Steckplatz für Arithmetikprozessor mit frei wählbarer Taktfrequenz**
- **Eingebauter Hi-Speed Festplattencontroller mit Autoboot zum Anschluß der Quantum-Festplatten. Direkter Zugriff des 68030 Prozessors auf diesen Controller ermöglicht optimale Datenübertragungsraten bis 4 Megabytes pro Sekunde!**

## Fordern Sie noch heute den ausführlichen GVP Prospekt an!

## AMIGA WORLD jetzt abonnieren!

Reservieren Sie sich jetzt ein Abonnement der meistgelesenen US-Amiga Zeitschrift. Brandheiße Informationen direkt von der Quelle. Bezugspreise incl. Versandkosten:

| | |
|---|---|
| Einzelexemplar | 13,- DM |
| Jahresabo 12 x 12,- = | 144,- DM |

## KOSTENLOSE INFORMATIONEN 6/89

senden wir Ihnen gerne zu allen Neuheiten und wenn Sie meinen, dies sei alles, bekommen Sie noch unsere Gesamtliste mit über 600 Artikeln. Einfach diesen Coupon auf eine Postkarte kleben und an DTM bzw. in der Schweiz an MICROTRON senden.

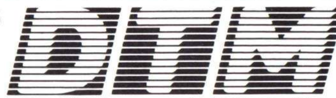
O Senden Sie mir Ihre Gesamtliste AMIGA Soft- und Hardware.
O Senden Sie mir detaillierte Informationen zu folgenden Produkten:

Alle unsere Produkte erhalten Sie in gutsortierten Fachgeschäften, die wir Ihnen gerne nennen sowie unseren DTM-Computershops in Wiesbaden. Selbstverständlich können Sie alle Produkte auch direkt schriftlich oder telefonisch bei DTM bzw. MICROTRON bestellen.

Alle Preise sind unverbindlich empfohlene Verkaufspreise. Preisänderungen und Irrtümer vorbehalten. Es gelten unsere AGB.

MICROTRON COMPUTERPRODUKTE
Postfach 69 Bahnhofstr. 2
Tel. 032 872429 Fax 032 872482
CH-2542 PIETERLEN

Poststraße 25
6200 Wiesbaden-Bierstadt
(0 61 21) 56 00 84
fax (0 61 21) 56 36 43




Werbung und EDV GmbH

# Im Land der Superlative?

Im April fand eine der umfangreichsten Computermessen, die COMDEX Spring '89, in Chicago statt. Chicago, die Stadt direkt am Michigan See gelegen, prahlt mit dem höchsten Gebäude, dem Sears Tower, und dem frequentiertesten Flughafen der Welt. Riesige Straßenkreuzer und ein scheinbar unendlich weites Land geben mir den Eindruck, als müßte hier einfach alles größer, höher und besser sein, doch was die Amerikaner unter einer der größten Computermessen verstehen, enttäuschte mich doch sehr, vor allem, wenn man kurz zuvor die CeBIT in Hannover besucht hatte. Typisch amerikanisch sind die beiden Gebäudekomplexe, in denen die neuste Hard- und Software gezeigt wird, mit so weichem und gepolsterten Teppichboden ausgelegt, daß man das Gefühl bekommt, über die Stände zu schweben.

Der Commodore-Stand fiel so klein aus, daß man ihn leicht übersehen hätte können, wenn da nicht das Commodore Logo an der Hallendecke gut sichtbar befestigt gewesen wäre. Der Stand war dann auch so leer, daß man sich innerhalb von drei Minuten einen kompletten Überblick verschaffen konnte. Desktop Publishing und Desktop Video waren die Schwerpunkte für AMIGA-Besitzer. Schließlich wurden noch Anwendungen im Bereich Musik, Büro und für zu Hause gezeigt. An Neuigkeiten war nichts zu finden, was nicht schon in Hannover gezeigt worden wäre. Auffällig war, daß Commodore nicht ein einziges Spiel vorführte, lediglich an einer Standwand waren einige Verpackungen befestigt. Möchte Commodore vom Spielcomputer-Image abrücken?

Aber nicht nur der AMIGA hat es schwer in den USA, auch bei der Konkurrenz, z.B. am ATARI-Stand, war Flaute angesagt, so daß einige Aussteller vorzeitig das Geschehen verließen oder erst gar nicht erschienen sind.

In Amerika haben IBM und Hersteller von kompatiblen Rechnern das Sagen. Nicht zu vergessen ist Apple mit dem Macintosh, der sich vor allem durch massiven Einsatz an fast allen Universitäten einen Namen gemacht hat.

Bei AMIGA und ATARI ST kann ich mich des Eindrucks nicht erwehren, daß die Geräte zwar in den USA entwickelt wurden, produziert werden sie aber in Taiwan oder Korea, und verkauft werden sie größtenteils in Europa und vor allem in Deutschland.

Ihr

Uwe Bärtels

# INHALT

## AKTUELLES



## SOFTWARE



## HARDWARE



## GRUNDLAGEN



## *FULL SPEED*

**Großer Assembler-Vergleich**

Die schnellste Programmiersprache ist mit Sicherheit Assembler. Doch die Auswahl dabei ist enorm. Welcher für Sie der richtige ist, welche Vor- bzw. Nachteile die einzelnen Assembler besitzen, welche Unterschiede existieren, welchen Befehlsumfang sie haben uvm. lesen Sie im großen Assembler-Vergleichstest ab Seite

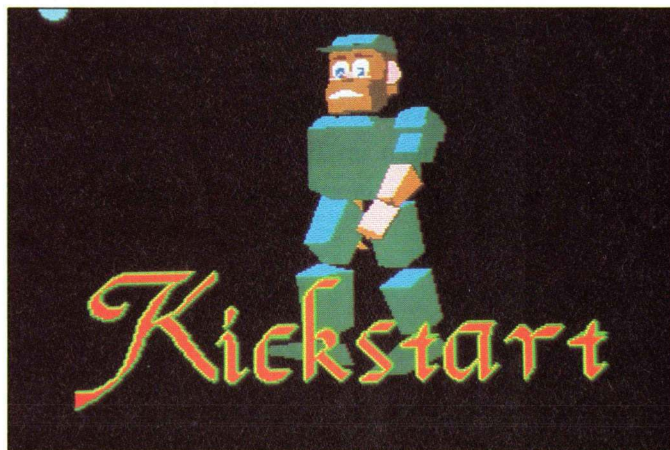
19



## ZOETROPE

ZOETROPE ist ein Gerät aus den Kindertagen der Bildanimation, dessen Geschichte weit in das letzte Jahrhundert zurückreicht. Das amerikanische Softwarehaus Antic hat den Namen aufgegriffen und ihr neuestes Animationsprogramm so getauft. Die zahlreichen Features des Programms ließen einiges erwarten; was schließlich und endlich dabei herauskam, lesen Sie ab Seite

74



# 6 '89

## PageStream

**Neuer DTP-Künstler**

PageStream nennt sich das neueste Desktop Publishing-Programm. Das in 100% Assembler geschriebene Programm soll neue Maßstäbe in der Ausdruckqualität setzen; egal, welchen Drucker sie besitzen, PageStream soll den passenden und angepaßten Treiber haben. Doch nicht nur in der Druckqualität sollen die Stärken des Programms liegen. Ob die Vorschußlorbeeren gerechtfertigt sind, lesen Sie ab Seite

28

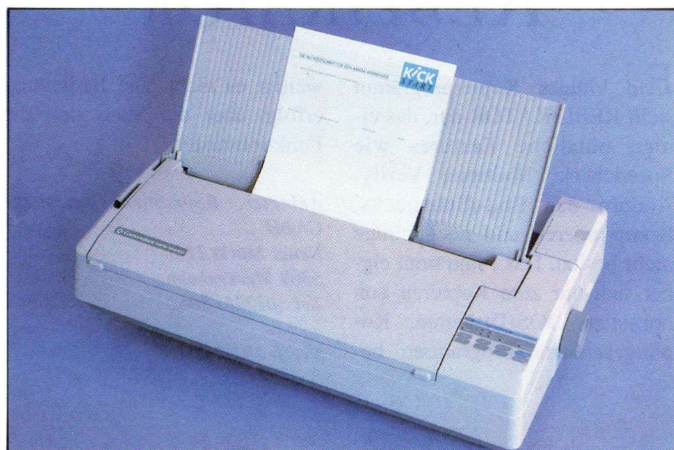



## MPS 1224 C

**Commodores neuer Drucker**

Ein DIN A3-Farbdrucker ist das neueste Peripheriegerät aus dem Hause Commodore. Der von Mannesmann hergestellte Drucker trägt jedoch die Bezeichnung Commodore MPS 1224 C. Wir testeten den Drucker auf Herz und Nieren, was dabei herauskam, lesen Sie ab Seite

12



## LISTING



## KICKS FÜR INSIDER



## SPIELETIPS



## SPIELE



## RUBRIKEN

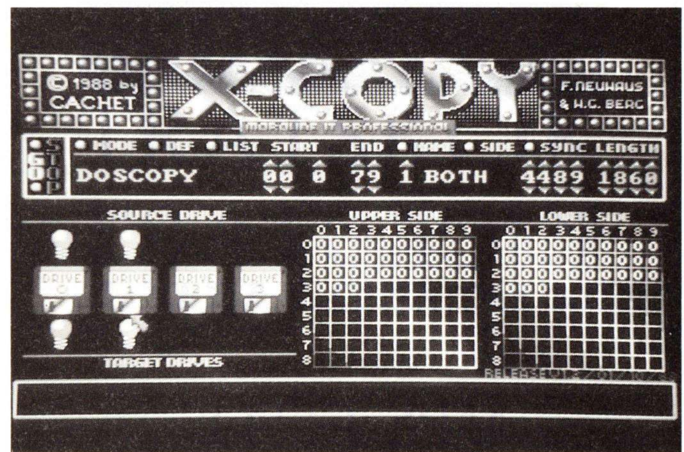
# NEWS

KICKSTART 6'89

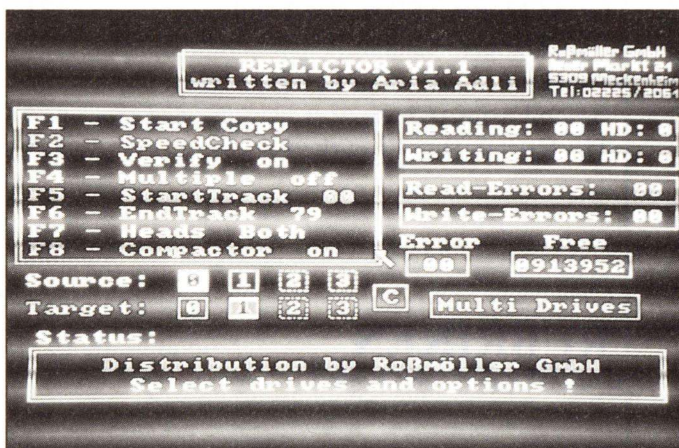
## *X-COPY UPDATE*

Kaum ein Kopierprogramm wurde so oft verkauft wie X-COPY von CACHET, diente es doch als hervorragendes Werkzeug zum Erstellen von Sicherheitskopien. Als eine Eigenart des AMIGA ist die Tatsache zu werten, daß er sogenannte Longtracks lesen aber nicht schreiben kann. Viele Softwarehäuser machten sich dies zu eigen und schützten so ihre Programme. X-COPY scheiterte so mit immer sobald sich ein Longtrack auf einer Diskette befand. Die neue Version ist jetzt in der Lage, auch die Longtracks zu kopieren. Eine kleine Hardwareerweiterung macht es möglich, sie wird an der Scartbuchse zwischengeschaltet und beinhaltet einen Oszillator, der die Taktfrequenz des AMIGA heraufsetzt. Das Ergebnis: Der AMIGA kann mehr Daten in derselben Zeit schreiben, somit auch Longtracks. Der kleine Hardwarezusatz soll zu einem konkurrenzlos günstigen Preis, im Rahmen eines Update-Services, auf der Markt kommen, so ließ zumindest CACHET verlauten.

*Anbieter: CACHET*
*Ostendstr. 32*
*7524 Östringen*




X-COPY ist jetzt in der Lage auch überlange Tracks zu kopieren - ein kleiner Hardwarezusatz macht's möglich.




Die Bedienung des Kopierprogramms erfolgt über die Maus und die Funktionstasten.

## *REBLIKATOR*

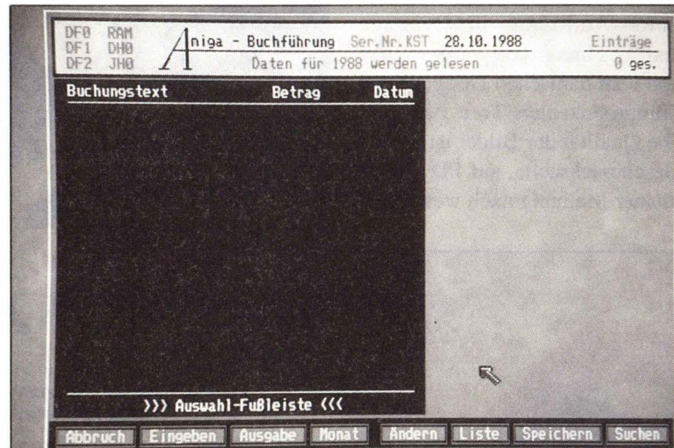
Ein billiges Kopierprogramm stellt REBLIKATOR dar, das einige nützliche Features wie Speedcheck, Multiple, Verify, Ändern des End-/Starttracks, Komprimieren und noch einige mehr besitzt. Das Programm eignet sich nur zum Kopieren von normalen DOS-Disketten. Kopiergeschützte Programme bewältigt es nicht. Die Bedienung erfolgt über die Maus und die Funktionstasten.

*Anbieter: Roßmöller Handshake GmbH*
*Neuer Markt 21*
*5309 Meckenheim*
*Tel. (02225) 2061*

*Preis: 29.95 DM*

# AMIGA HAUSHALTSBUCHHALTUNG

Das Programm ist für die private Haushaltsbuchhaltung konzipiert. Der Preis von lediglich 29.95 DM ist erfreulich, ein Updateservice ist darin enthalten. Die Bedienung des Programms ist voll mausgesteuert. Wichtige Menüpunkte sind außerdem noch mit Tastatursequenzen belegt. Die Programmführung ist in deutscher Sprache, die Dokumentation, die lediglich auf Diskette vorliegt, ebenfalls. Ein Manko des Programms besteht darin, daß mindestens 1 MB RAM zur Verfügung stehen muß. AMIGA-Besitzer mit nur 512 KB haben das Nachsehen.




Die Aufmachung der Haushaltsbuchhaltung ist recht farbenfroh.

Gegenüber älteren Versionen des Programms AMIGA-Haushaltsbuchhaltung ist das Programm nicht mehr in BASIC geschrieben und hat somit eine erhebliche Geschwindigkeitssteigerung erfahren.

*Anbieter: 3 S Service*
*Herrmanns & Kommelter*
*Vom-Bruck-Platz 45*
*4150 Krefeld*
*Tel. (02151) 399833*

*Preis: 29.95 DM*

# K-DATA

K-DATA nennt sich eine neue Datenbank aus dem Hause KUMA Computers Ltd. hauptsächlich bekannt durch den Seka-Assembler. Im Gegensatz zu diesem Assembler findet die Benutzerführung von K-Data unter Intuition statt. Eine leichtere Bedienung ist somit gewährleistet. Alle typischen Datenbank-Features sind implementiert. Die Länge des Datenfiles ist lediglich durch das Speichermedium begrenzt, bei Bedarf werden Daten nachgeladen und gespeichert. Der Verkaufspreis von K-DATA liegt bei ca 170.- DM.

*Anbieter: Fachhändler*

# ANTI-VIRUS,DIE...

Im Zuge der Zeit kommt diesmal ein Anti-Virus-Programm aus England. Hauptmerkmal des Programms ist die Möglichkeit, eine Virus-Datenbank anzulegen. Ansonsten bietet es nichts Weltbewegendes. Bessere und leistungsfähigere Programme existieren zahlreich in den diversen PD-Sammlungen. Das Programm lautet schlicht Virus und hat nichts mit dem gleichnamigen Spiel zu tun.

# TOWER AMIGA

In Ausgabe 4/89 der KICKSTART testeten wir den TOWER AMIGA ausführlich, ab sofort wird er mit einem verlängerten Maus- und Tastaturkabel ausgeliefert. Auch besitzt die TOWER-Tastatur jetzt einige Leuchtdioden, die den Betrieb der Festplatte und der Laufwerke anzeigen. Der Preis des TOWER AMIGA bleibt gleich.

*Anbieter: CSS*
*Auf der Warte 46*
*6367 Karben 1*
*Tel. (06039) 57756*

# UNIVERSAL SCANNER

Die Firma PRINT TECHNIK hat ihren Preis für den UNIVERSAL SCANNER auf 1198.- DM gesenkt. Des weiteren besitzt der Scanner jetzt eine neue Software-Fassung, die erlaubt, das Gerät als Drucker zu verwenden. Der Scanner kann als Thermodrucker mit einer Epson-Druckeremulation betrieben werden. Die Druckqualität kann sich durchaus sehen lassen. Auch wurde das globale Konzept der Software ein wenig optimiert und geändert. Ein neues Handbuch befindet sich z.Z. noch in Arbeit, wird aber demnächst zur Verfügung stehen. Ein ausführlicher Test des Scanners war in Ausgabe 11/88 der KICKSTART zu lesen.

*Anbieter: PRINT TECHNIK*
*Computer Peripherie GmbH*
*Nikolaistr. 2*
*8000 München 40*
*Tel. (089) 368197*

*Preis: 1198.- DM*

# Commodore Jubiläumskampagne

Laut Commodore wurden über 1 Million AMIGAs weltweit verkauft! Aus diesem Grunde veranstaltet Commodore eine Jubiläumskampagne, die aus einer breit angelegten Promotion-Aktion besteht. 30 Promotoren in 200 Kaufhäusern unterstützen den Verkauf des AMIGA 500. Als Jubiläums-Bonbon für den AMIGA 500-Neukunden werden 4 Software-Programme seiner Wahl (aus einem Angebot von 6 Programmen) beigelegt. Darunter findet sich eine Textverarbeitung, ein Grafik-, Erdkunde- und Mathematikprogramm, das Actionspiel Katakis und das Strategiespiel Jeanne d'Arc.

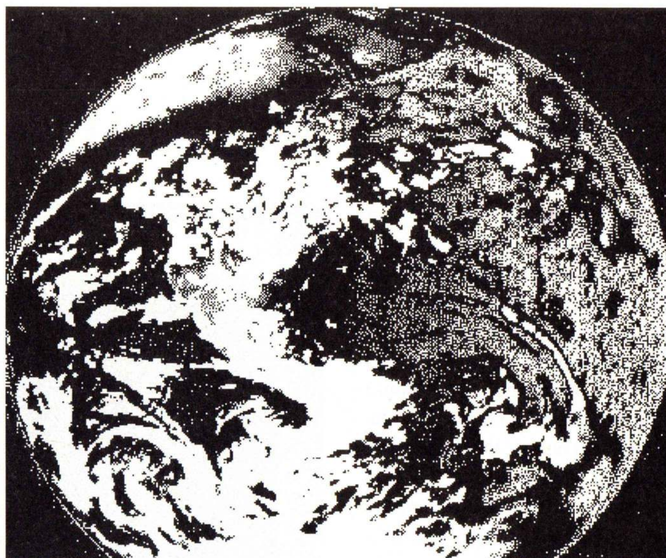
# MEDIA LINE

Die amerikanische Firma FREE SPIRIT SOFTWARE Inc. hat mit der Reihe MEDIA LINE eine Sammlung von Bildern, Grafiken und Fonts herausgebracht, die von verschiedenen Anwendungsprogrammen genutzt werden können.



Auf den ANIMATION Background-Disketten finden sich 32 verschiedene Bilder.

Für Video- und Animationsbackgrounds sind zwei Disketten vollgepackt mit Bildern, die mit Aegis Video Titler, Light, Camera, Action, Turbo Silver und Sculpt 3D bzw. 4D weiterverarbeitet werden können. Die 30 Hintergrundbilder der zwei Disketten sind in vier unterschiedliche Rubriken unterteilt. Alle Bilder liegen in der Interlace-LoRes-Auflösung in Overscan und 32 Farben vor und stammen von dem dänischen Designer und Filmproduzenten Tore Bahnson. Die Qualität der Bilder ist jedoch Geschmackssache, auf PD gibt es meiner Meinung nach wesentlich ansprechendere Bilder und Grafiken zu einem wesentlich günstigeren Preis. Immerhin schlagen die Bilder mit ca 70.- DM zu Buche. Des weiteren stehen unter dem Titel MEDIA LINE AMIGA CLIPART kleinere Grafiken für DesktopPublishing-Programme zur Verfügung. Da die Grafiken im IFF-Format vorliegen, können sie von fast allen Zeichenprogrammen nachbearbeitet werden. Der HiRes-Interlace-Modus garantiert detailreiche Grafiken. Der Preis der MEDIA LINE CLIPART DISK 1 liegt bei ca 60.- DM. Neben diesen beiden MEDIA LINE-Sammlungen sind unter derselben Reihe noch zwei Font-Disketten zu beziehen, die jeweils 9 Bit-Map-Fonts beinhalten. Diverse Animationsprogramme können auf die Fonts zurückgreifen.

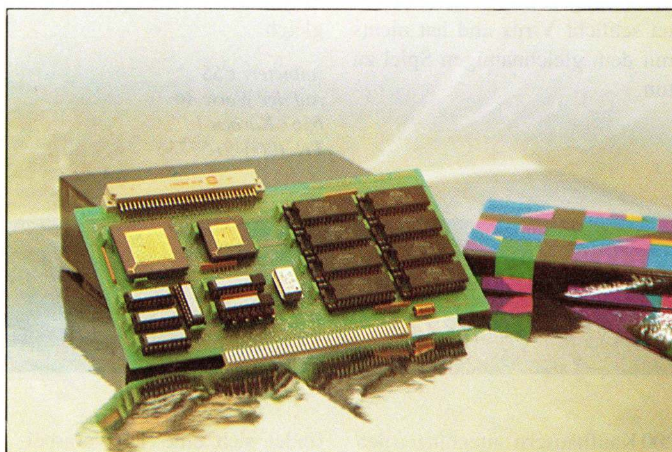


Dür Desktop Publishing-Programme können die zahlreichen Grafiken der Clipart-Diskette herangezogen werden.

*Anbieter: Fachhändler*

# ANIMATE TURBO BOARD-FAMILIE

HARMS Computer-Systeme aus Bremen vertreibt eine Turbo Board-Familie, die alle den 68020-Prozessor mit 68881 (68882) als Grundkonfiguration beinhalten. Das Animate Turbo Board I ist für den AMIGA 1000 bzw. 500 konzipiert (siehe Test in Ausgabe 4/89). Dieses Board stellt die preisgünstigste Lösung dar. Das Turbo Board II besitzt gegenüber dem Board I die Möglichkeit, auf den 68000-Prozessor zurückzuschalten. Die Kompatibilität bleibt somit gewährleistet. Dieses Board ist für den AMIGA 2000 und 1000. Das Animate Turbo Board III ist ausschließlich für den AMIGA 2000 konzipiert. Es wird in den MMU-Slot des A2000 gesteckt. Der mathematische Coprozessor kann mit bis zu 32 MHz getaktet werden. Außerdem befindet sich auf der Karte ein 32 Bit Static-RAM. Das RAM läßt sich von 128 KB auf 1 MB erweitern. Als besonderes Feature besitzt die Karte einen 32 Bit Peripherie-Adapter zum Anschluß von zusätzlicher Hardware. Beispielsweise können eine weitere 32 Bit-RAM-Karte oder eine Multifunktionkarte angesteckt werden. Laut Hersteller sind die Karten zu allen gängigen Animationsprogrammen kompatibel (Sculpt 3D, Animate 4D, Turbo Silver, Modeler 3D, Video Scape, Video Titler, Video Effects 3D, Deluxe Video und viele mehr).



Das Animate Turbo Board III ist für den AMIGA 2000 konzipiert und besitzt einen 32 Bit Peripherie-Adapter.

*Anbieter:*
*HARMS Computer-Systeme*
*Harselfeldstr. 18*
*2800 Bremen 1*
*Tel. (0421) 444790*




...UND

PLÖTZLICH

**AMIGA**

GIBT'S

FÜR DICH

EINE

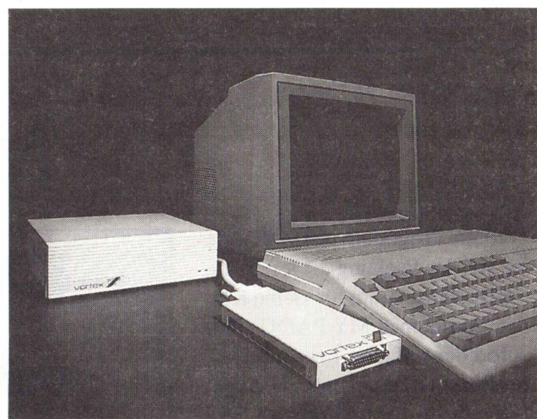
AMIGANTISCHE

**FEST-**

**PLATTE**

## Die neue vortex-Festplatte System 2000 für AMIGA 500 und AMIGA 1000. Serienmäßig autobootfähig ab KICKSTART 1.2. Zum sensationellen Preis von DM 998,–.*

* unverbindliche Preisempfehlung für 20 MB




Ja, mein lieber Freund, AMIGA, die neue vortex-Festplatte SYSTEM 2000 für AMIGA 500/1000 wird dir Beine machen: Kapazitätsmäßig habe ich jetzt ganz locker 20, 30, 40 oder 60 MB zur Verfügung. Und die mittlere Zugriffszeit, Junge, Junge: Ganze 30 ms bei der 60 MB-Version.

Der absolute Überhammer: Modernste Chip-Technologie (vortex DMA-Gate-Array) ermöglicht eine Übertragungsrate von 1 MB/sec.!!! Whowww!

**Bei diesen vortex-Fachhändlern gibt's ab sofort die amigantische Festplatte:**

**1000:** Mükra, Berlin 42; Schlichting, Berlin 61; MiniSoft's DFÜ Shop, Berlin 62; Karstadt, Berlin 65.
**2000:** Brinkmann, 2000 Hamburg; Lavorenz, 2082 Uetersen; Ihlow&Kruse, 2100 Hamburg 90; Andreas Voss, 2280 Westerland/Sylt; MCC, 2300 Kiel; Brinkmann, 2800 Bremen; Dodenhof, 2802 Ottersberg-Posthausen.
**3000:** Com Data, 3000 Hannover; Ludwig Haupt, 3100 Celle; Computer Studio Frank Ueckert, 3180 Wolfsburg 11; Witte Bürotechnik, 3250 Hameln.
**4000:** Data Becker, 4000 Düsseldorf; OCB, 4422 Ahaus; Delo Computer, 4600 Dortmund; Compimate, 4800 Bielefeld; MC-Byte, 4830 Gütersloh; Knicker & Wortmann, 4950 Minden.
**5000:** AB Computer, 5000 Köln; Kaurisch, 5500 Trier; Wrede Bürotechnik, 5778 Meschede.
**6000:** Computer Transparent, 6070 Langen; Herbig, 6120 Erbach; Computer Wolf, 6200 Wiesbaden; GTI, 6370 Oberursel; Shop 64, 6680 Neunkirchen/Saar.
**7000:** Schreiber Computer, 3×in 7000 Stuttgart, 7032 Sindelfingen, 7100 Heilbronn, 7140 Ludwigsburg, 7250 Leonberg und 7530 Pforzheim; Rothfuß KG, 7142 Marbach; Wiebelt, 7730 Villingen-Schwenningen.
**8000:** Seemüller, 8000 München 2; Media Markt, 8000 München 45, 8070 Ingolstadt, 8300 Landshut, 8400 Regensburg, 8580 Bayreuth und 8605 Bamberg; BSC, 8000 München; Elektro Egger, 8000 München 60; COM, 8000 München 80; Promarkt, 8032 Gräfelfing; Horten, 8500 Nürnberg; Tevi Markt, 8500 Nürnberg 80; Elektro Stender, 8640 Kronach; Top 3 Markt, 8700 Würzburg; Bissinger, 8833 Gundelfingen.

**vortex**
COMPUTERSYSTEME

...UND PLÖTZLICH LEISTET IHR COMPUTER MEHR

# FONTS FÜR SCULPT 3D UND ANIMATE 4D



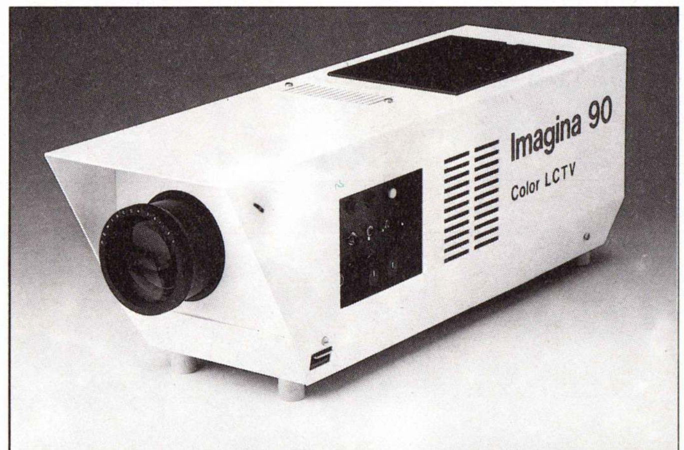
Der LCD-Font, verwendet im Programm Animate 4D.

Die amerikanische Firma BYTE by BYTE, bekannt durch die Programme Sculpt 3D und Animate 4D, hat für diese beiden Programme eine Zusatzdiskette mit drei Fonts herausgebracht. Die Fonts BEVEL, LCD und PREMIERE liegen im 3D-Format vor und können mit den oben genannten Programmen weiterverarbeitet werden. Jeder, der schon einmal versucht hat, Fonts zu erstellen, weiß, wie umständlich die ganze Sache ist. Die drei Fonts finden sich auf insgesamt zwei Disketten.

# DER UNIVERSELLE BILDSCHIRM

Wer den AMIGA-Bildschirm 'unbegrenzt' groß haben möchte, ist mit dem IMAGINE 90 bestens bedient. Das Gerät wirft ähnlich einem Diaprojektor das Bild auf eine Leinwand. Die Größe der Projektion ist auf maximal 3 Meter in der Diagonalen begrenzt. Besonders geeignet ist das Gerät für Werbung und Präsentationen. Der Anschluß des AMIGA an das Gerät erfolgt über die Video-/Audio-Schnittstelle des IMAGINE 90. Ein UHF/VHF-Tuner ermöglicht das Empfangen von Fernsehprogrammen.

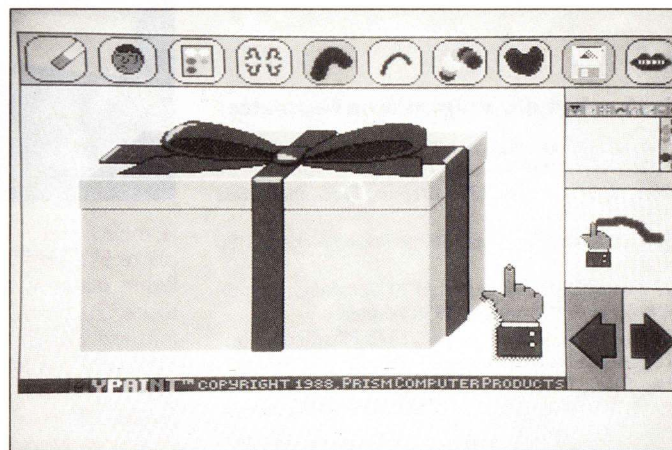
*Anbieter: VIA Computer GmbH*
*Apostelnstr. 9-11*
*5000 Köln 1*
*Tel. (0221) 210091*




IMAGINE 90 ermöglicht eine Bilddarstellung von bis zu 3 Metern in der Diagonalen.

# MY PAINT

"A Paint Programm just for Kids!" ist auf der Packung zu lesen, ein Zeichenprogramm für kleine Kinder. Startet man MY PAINT, fällt sofort die einfache, naive und farbenfrohe Aufmachung ins Auge. Der obere Bildschirmrand beinhaltet 10 verschiedene Symbole, die die verschiedenen Modi des Programms bestimmen. Am rechten Bildschirmrand kann die Zeichenfarbe gewählt werden. Alle Anwählfelder sind animiert. Auf der Diskette finden sich 28 Bilder, die von den kleinen Künstlern nachbearbeitet werden können, wobei jedes Bild einen eigenen gesampelten Sound besitzt. Wird beispielsweise das Dinosaurierbild geladen, ist aus dem Lautsprecher der Name des Bildes zu entnehmen. Leider geschieht das in englischer Sprache, so daß es für den deutschsprachigen Raum wenig Nutzen bringt. Selbiges gilt auch für die Dokumentation, die zwar einfach und mit vielen Illustrationen versehen, aber in englischer Sprache ist. CASABLANCA, deutscher Vertreiber, ließ aber verkünden, daß hier Abhilfe geschaffen und der Sound verändert wird, ebenfalls soll das Programm in einer PAL-Version betrieben werden. KINDER können mit dem Malprogramm MY PAINT spielend den Umgang mit dem Computer lernen und erste Schritte mit Maus und Piktogrammen üben.




MY PAINT ist speziell für Kleinkinder geschrieben worden.

*Anbieter: CASABLANCA*
*Nehringskamp 9*
*4630 Bochum*
*Tel. (0234) 411994*

*Preis: 89,- DM*

# AMIGA MIT 1 MB-CHIPMEM

Es ist soweit, der AMIGA 2000 wird mit 1 MB-Chipmem ausgeliefert. Den AMIGA 500 mit 512 KB. Der Käufer kommt also zunächst nicht in den Genuß von 1 MB-Chipmem. Wer jetzt jedoch meint, es handle sich um das bekannte ECS (Enhanced Chip Set), der hat sich geirrt. In den neuen Rechnern findet sich lediglich der neue Fat Agnus, der eben seinen Adressierungsbereich auf 1 MB ausgeweitet hat. Der Vorteil liegt klar auf der Hand, Mal-, Zeichen-, Sound- und etliche Programme mehr werden die 1 MB-Chipmem zu schätzen wissen. Jedoch besitzt dieser AMIGA 2000 (AMIGA 500) noch einen kleinen Haken. Wird ein Reset ausgelöst, wird ein Kaltstart vorgenommen. Resetfeste Programme verschwinden aus dem Speicher, resetfeste RAM-Discs werden unweigerlich gelöscht. Der Vorteil eines Kaltstarts besteht jedoch in der Tatsache, daß Viren ebenfalls keine Überlebenschance haben. Jedoch wird das Programm 'Setpatch' mitgeliefert, das dem Problem zuleibe rückt. Nach Aufruf des Programms führt der AMIGA wie gewohnt einen Warmstart aus. Aus diesem Grunde ist es anzuraten diesen Befehl in den verschiedenen 'Startup-sequenzen' der Disketten zu installieren, sofern Sie mit resetfesten Programmen arbeiten. In den neuen Rechnern ist ebenfalls das neue Betriebssystem V 1.3 installiert. Besitzer eines alten AMIGA 2000, die ihren Rechner nachrüsten wollen, müssen sich noch gedulden, erst in ca. 2 Monatenb plant Commodore einen Nachrüstsatz auf den Markt zu bringen. Allerdings haben AMIGA 1000-Besitzer keine Möglichkeit, den neuen Fat Agnus einzusetzen, selbiges gilt für AMIGA 2000 A-Besitzer. Ob eine Anpassung an diese Rechnertypen vorgenommen wird, steht z.Z. noch nicht fest.





VON MARKUS NERDING

# MPS 1224C
# Commodores neue Errungenschaft



*Nachdem die Druckerserie CBM_MPS2xxx, hinter der sich Geräte der Firma NEC verbargen, ausgelaufen ist, hat man sich bei Commodore nach einem neuen Partner umgesehen. Gefunden hat man ihn bei Mannesmann, die für Qualitätsdrucker für den Geschäftsbereich bekannt sind. Doch seit geraumer Zeit stellt man bei Mannesmann auch Low Cost-Geräte her. Worum handelt es sich also beim neuen MPS 1224C?*

## Bedienung

Schon an der Verpackung kann man erkennen, daß der MPS 1224C ein DIN A3-Drucker ist, der auch nur in diesem "Überformat" lieferbar ist. Zudem ist er nur in der Farbversion verfügbar, was den Preis von DM 1998.- erklärt. Hat man ihn aufgebaut und versucht das erste Mal, Endlospapier einzuspannen, beginnt allerdings schon wieder die Demontage. Als erstes versucht man, die Frontabdeckung (1) hochzuklappen und bemerkt erstaunt, daß man diese sogleich in der Hand hält. Wir ersparen uns an dieser Stelle einen Kommentar dieses Vorfalls, denn dieser Fehler ist laut Hersteller bereits erkannt und bei weiteren Versionen behoben. Wir entfernen die klappbare Einzelblattzuführung (12) und versuchen, die Traktorklappen (14) zu öffnen. Doch leider stößt man auf ein Hindernis in Form der Druckwalzenabdeckung (10). Nach deren Entfernen gelingt es endlich, die Traktorklappen zu öffnen, was in der Redaktion mit lautem Applaus bedacht wird. Erwähnt wurde dieser Vorgang nur, weil ich noch keinen Drucker gesehen habe, der in dieser Beziehung ähnlich umständlich zu bedienen ist. Es muß jedoch erwähnt werden, daß man diesen Aufwand nur selten betreiben muß, denn das Endlospapier wird von nun an nur selten herausgenommen werden. Es verbleibt bei einem Wechsel zu Einzelblättern in einer Parkposition, und nur, wenn das Endlospapier ausgeht oder ein Papierstau auftritt, muß die Prozedur wiederholt werden. Über das optische Erscheinungsbild kann man sich vielleicht streiten, aber es kann als schlicht, die Bedienungselemente als recht spartanisch bezeichnet werden. Das Bedienungsfeld ist klein und schlecht gegliedert. Ein Teil der Funktionen ist nur bei Online, andere bei Offline und weitere bei Kombinationen anwählbar. Eine Erklärung hierfür konnte nur mit viel Mühen gefunden werden (Rationalisierung?!). Es kam aber während des Tests zu einigen Fehlbedienungen. Es fällt auf, daß keine zusätzlichen Schriftarten anwählbar sind, und ein Blick ins Handbuch bestätigt diese Vermutung. Wer nicht wehmütig zur Konkurrenz blicken möchte, kann sich weitere Schriftarten in Form von Steckkarten kaufen und diese wahlweise in den (einzigen) Font-Schacht stecken.

Die Grundeinstellungen wie z.B. die Emulation, Papierlänge, rechter und linker Rand, alle Parameter der seriellen Schnittstelle, Zeichensatz (Epson, IBM, NEC und 12 nationale) und einiges mehr werden über ein Dialogmenü eingestellt. Der Drucker gibt hierzu eine Anzahl Möglichkeiten aus, und per Tastendruck wird mit dem Druckkopf ein Punkt angefahren. Auf Bestätigung wird dieser akzeptiert. Alles in allem eine sehr komfortable Möglichkeit, hardwaremäßig Druckereinstellungen vorzunehmen, die früher gar nicht oder nur mit DIP-Schaltern realisiert wurden.

Ein Blick unter die Haube ist als ein weiterer Lichtblick in diesem Test zu sehen, denn die Mechanik und der Druckkopf machen einen soliden und stabilen Eindruck. Die Farbbandkassette ist hingegen etwas klein geraten und wird deshalb häufiger gewechselt werden müssen. Zum Ausdruck von Text können jedoch schwarze 4-Zonen-Bänder verwendet werden, bei denen der Drucker automatisch umschaltet und somit die vierfache Bandlänge zur Verfügung hat.

Gut ist die Papierparkfunktion, die es erlaubt, das Endlospapier im Traktor aufzubewahren, während man mit Einzelblättern weiterdruckt. Der Einzelblatteinzug ist zwar einfach in Position zu bringen, doch der Umschalter Traktor/Friktion ist schwergängig, und die seitlichen Führungen sind sehr klein ausgefallen. Dadurch wird der ordnungsgemäße Einzug eines Blattes nicht unbedingt unterstützt, was sich jedoch in erster Linie bei Mehrfachformularen unangenehm auswirkt.

Für die Verwendung von Einzelblättern gibt es jedoch optional auch einen Zweischachteinzug, der einfach auf den Drucker aufgesteckt wird. Damit können dann Geschäftsbriefe mit Folgeseiten bedruckt werden, soweit es die verwendete Software zuläßt.

Bei der Verwendung von Endlospapier ist es im allgemeinen so, daß die Abrißkante nicht direkt genutzt werden kann, weil sich die Perforation noch ein gutes Stück unterhalb befindet. Die gewählte Lösung des Problems ist sehr komfortabel, denn man drückt eine Tastenkombination, die Perforation fährt bis zur Abrißkante - man reißt das Papier ab, drückt wieder die Tastenkombination und das Papier fährt an den Seitenanfang zurück - praktisch!

## Druck

Verlassen wir die Beurteilung der Handhabung und Mechanik, und wenden wir

```
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^
```

*Der Selbsttest bekennt Farbe*

uns dem eigentlichen Sinn eines Druckers zu - dem Ausdruck geistiger/-loser Machwerke. Erwartungsgemäß (schließlich ist er ein 24-Nadler) recht schnell ist der Ausdruck, und auch die Schriftqualität läßt nicht viel zu wünschen übrig. Schon im Selbsttest bekennt er kräftig Farbe, und auch bei den für AMIGA-Fans beliebten Hardcopies kommen für Matrixdrucker gute Ergebnisse zutage.

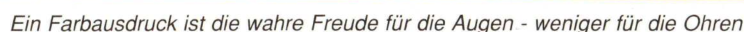
Für die Bewertung der Druckgeschwindigkeit zogen wir die Druckerdaten aus dem großen Vergleichstest in Heft 10/88 heran. Hier schneidet der MPS 1224C nicht sehr gut ab, er muß sich sogar zu den langsamsten Druckern zählen lassen. Erwartungsgemäß hoch sind die Zeiten zur Datenaufnahme ausgefallen, denn der Puffer beträgt nur 1KB [soll lt. Anbieter ab Mitte des Jahres auf 8 KB (anwählbar) erweitert werden], aber auch die Gesamtzeit für den Druck liegt recht hoch. Hier kann der Drucker in der Konkurrenz keine Lorbeeren ernten. Bleibt die Frage nach einer Einordnung der Daten und damit des Druckers.

## Fazit

Optisch (obwohl Geschmacksache) ist der Drucker schlicht und unauffällig, hier fällt lediglich seine Breite ins Auge.

Die Bedienung ist in den angegebenen Bereichen (Endlospapier, Tastenfeld) nicht besonders gut gelöst, allerdings hat man für einige Punkte bereits Abhilfe geschaffen und den Drucker diesbezüglich überarbeitet. Gut gelöst ist die Papierabrißfunktion, die das Papier für einen sauberen Abriß vor und danach wieder in die Ausgangsstellung transportiert.

Auch das Dialog-Menü für die vielfältigen Druckereinstellungen dient der gu-



*Ein Farbausdruck ist die wahre Freude für die Augen - weniger für die Ohren*

ten Bedienbarkeit und ermöglicht es auch dem Laien einfach Einstellungen am Drucker vorzunehmen. Technisch macht er einen soliden Eindruck (Druckkopf, Kopfführung); zu bemängeln ist hier der schwergängige Umschalter (Endlos-/Einzelblattpapier) und die (noch) lose Frontabdeckung. Die Druckgeschwindigkeit ist nicht besonders hoch, doch ist die Qualität dank des 24-Nadeldruckkopfs sehr gut.

Farbdruck, Emulationen (Epson, NEC, IBM), Zeichensätze, zwei Schnittstellen (seriell und parallel) und das breite Format sind die wichtigsten Optionen. Man vermißt hier hauptsächlich weitere Schriftarten (Fonts) und einen größeren Druckerpuffer.

Ob man die einzelnen Funktionen benötigt oder nicht, muß man sich vor einem Druckerkauf überlegen. Der MPS 1224C ist ziemlich universell und komplett ausgestattet und kann sich dadurch preislich sehen lassen. Über die Gewichtung der Druckgeschwindigkeit und einiger Bedienungspunkte muß man sich im klaren sein, und der Heimanwender muß sich vermutlich etwas Platz auf dem Tisch schaffen, um den Burschen unterzubringen.

■

## Kurzkritik

+ Farbdruck
+ gute Mechanik
+ komfortable Dialog-Menüs
+ vielfältige Einstellungen
+ emuliert NEC, IBM und EPSON

- teilweise umständliche Bedienung
- nur eine Schriftart
- Druckerpuffer zu klein (1 kB)

| | MPS 1224C | | NEC P6 PLUS | |
|---|---|---|---|---|
| Geschäftsbrief: DIN 32751 | | 40:00 | | 31:00 |
| NLQ 10 cpi: (33K Datei) | 10:26 | 10:46 | 0:18 | 8:36 |
| DRAFT 12 cpi: (wie oben) | 4:49 | 4:57 | 0:19 | 3:39 |
| Grafik | 0:45 | 0:47 | 0:17 | 0:18 |

Druckgeschwindigkeiten [Zeit zur Datenaufnahme (erste Spalte) und Gesamtzeit für Druck (zweite Spalte) (die Daten sind nicht direkt mit denen aus dem Test in Heft 10/88 zu vergleichen, weil die Rechner verschieden konfiguriert waren)

## CBM MPS 1224C

Ausstattung:

| | |
|---|---|
| Schnittstelle | seriell/parallel |
| Speicher | 1 kB |
| Emulationen | Epson LQ2500<br>NEC Pinwriter<br>Proprinter XL24 |
| Font | Courier |
| Fontgröße | 10, 12, 15, 17, 20, proportional |
| Druckqualität | Draft/LQ |
| Auflösung | 360x360 dpi |
| Geschwindigkeit | 220 cps (Draft 10cpi)<br>72 cps (NLQ 10cpi) |
| Papiertransport | Schub/Friktion |
| Papierparkfunktion | ja |
| Max. Papierbreite | 410 mm (DIN A3) |
| Farbdruck | ja |
| Handbuch | deutsch |
| Einstellung | gedruckte Menüs |
| Sonstiges | HEX-Dump<br>Download-Zeichengenerator |

Preis: DM 1998.- bis DM 2198.-

Optional: Fontkarten (Courier (?), Italic, Gothic), Einzelblatteinzug (Ein/Zweischacht)

Anbieter: Fachhandel
CDC, Bad Homburg,
Tel.: 06172 - 2 47 48

# *European Computer* TRADESHOW

*Obwohl zum ersten Mal veranstaltet, war sie für Spielefreaks sicherlich die bisher interessanteste Messe des Jahres. Allerdings, öffentlich war die ECTS nicht, und geladen hatte man nur Händler und Presse.*

Entsprechend dezent waren die Stände aufgebaut.Schlicht, funktional das Design. Besonders angenehm aufgefallen ist mir die Abwesenheit des Lärms der PCW-Präsentationen und des Gestanks noch nicht getrockneter Farbe und des Aufbautenkleisters.Anwesend war eigentlich nur der "harte Kern". Gesichter, die man als Fachjournalist auf jeder Messe sieht; Jonathan Ellis (Psygnosis), Bob Jacob (Cinemaware), John Symes (Microdeal), Lesley Mansford (Electronic Arts), John Forrest (Ubisoft) und so weiter. Trotzdem war es diesmal ein wenig anders als sonst. Das fehlende Gedränge ließ Raum für Gespräche, und der familiäre Rahmen war sicherlich geeignet, bereits bestehende Kontakte auszubauen. Entsprechend gab es nicht nur Produkte zu sehen, sondern auch Interna zu hören.

Interessant ist ganz sicherlich, daß Telecomsoft (Rainbird, Firebird) mittlerweile an MicroProse verkauft wurde: Michtron bei einem Deal mit EAS (Ringside) reingefallen ist (die geknackte Version war bereits überall in den USA, bevor EAS in der Lage war, an Michtron ein funktionierendes Master zu liefern Digital Magic Software mit dem Spiel Scorpion zuerst bei Gainstar, dann bei Gold Disc und mittlerweile bei einem weiteren Softwarehaus auf der Matte stand, jeweils mit der Aussage, man hätte mit "den anderen" nur eine lose Absprache getroffen; der Verkauf von AMIGA-Spielen im Durchschnitt pro Produkt in den letzten 18 Monaten um 60% zurückgegangen ist - d.h. wurden vor eineinhalb Jahren von einem Spiel 10.000 Exemplare verkauft, so sind es jetzt gerade noch 4000 im Durchschnitt.

In diesem Zusammenhang wird auch verständlich, daß mindestens fünf namhafte US-Softwarehäuser in finanziellen Schwierigkeiten sind und Rainbow Arts den Vertrag mit US Gold wegen unbefriedigender Verkaufszahlen gekündigt hat.


Imposant sieht das Eingangsportal der ECTS aus.

Sicher ist manche Information nicht verifizierbar und mit Skepsis zu betrachten, wie beispielsweise die Aussage, daß Teleconsoft verkauft wurde, weil British Telecom über

# KICK ME, AMIGA

# R.C.T.

KICK

Requester und Menü Construction Tool für den AMIGA

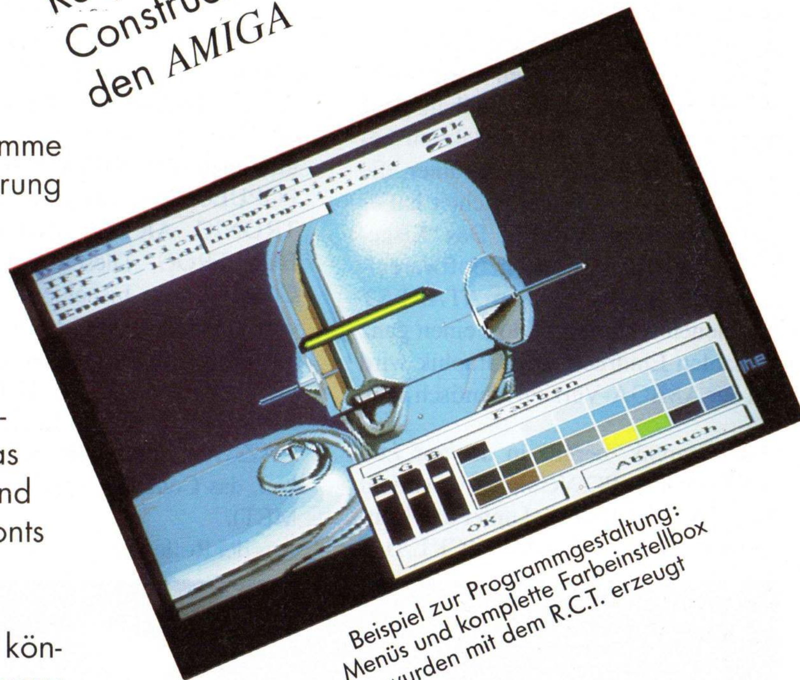
## Der Weg in völlig neue Dimensionen

Wollten Sie nicht schon immer ihre Programme mit einer professionellen Benutzerführung versehen?

Mit dem R.C.T. ist das kein Problem, egal welche Programmiersprache Sie wählen.

Das Erstellen einer grafikorientierten Oberfläche ist für viele ein Buch mit sieben Siegeln. Das R.C.T. hilft Ihnen, Requester mit Gadgets und Images oder Menüs in verschiedenen Schriftfonts zu konstruieren.

Die konstruierten Requester und Menüleisten können mit den Funktionen der mitgelieferten Library verwaltet werden. Die Library kann von allen Programmiersprachen genutzt werden, auch von Basic aus.
Bei Programmiersprachen wie C oder Assembler besteht die Möglichkeit, die Funktionen der Library direkt an das Programm zu linken.
Der Sourcecode-Generator ermöglicht die Erzeugung von C-, Assembler- und GFA-BASIC-Source von den konstruierten Requestern und Menüleisten.



Beispiel zur Programmgestaltung: Menüs und komplette Farbeinstellbox wurden mit dem R.C.T. erzeugt

**R.C.T. DM 129.-**




### Bestellcoupon

Bitte senden Sie mir:

☐ R.C.T. DM 129.-

zuzüglich Versandkosten:
Inland DM 7.50
Ausland DM 10.-

Auslandsbestellungen nur gegen Vorauskasse
Bei Nachnahme zuzüglich DM 3.80 Nachnahmegebühr

Name

Straße

Ort

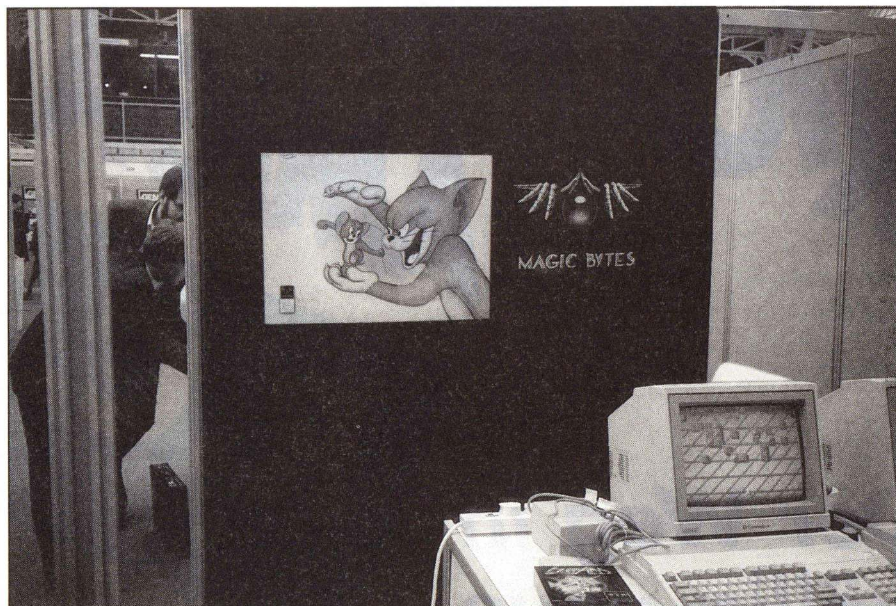
Datum Unterschrift

MAXON-Computer / Industriestr. 26 / 6236 Eschborn

fünf Mio. Pfund zugesetzt haben soll. Aber Tatsache ist, daß wirklich gute Umsätze in der Branche nur von ganz wenig Softwarehäusern und nur mit einzelnen Produkten erzielt werden.

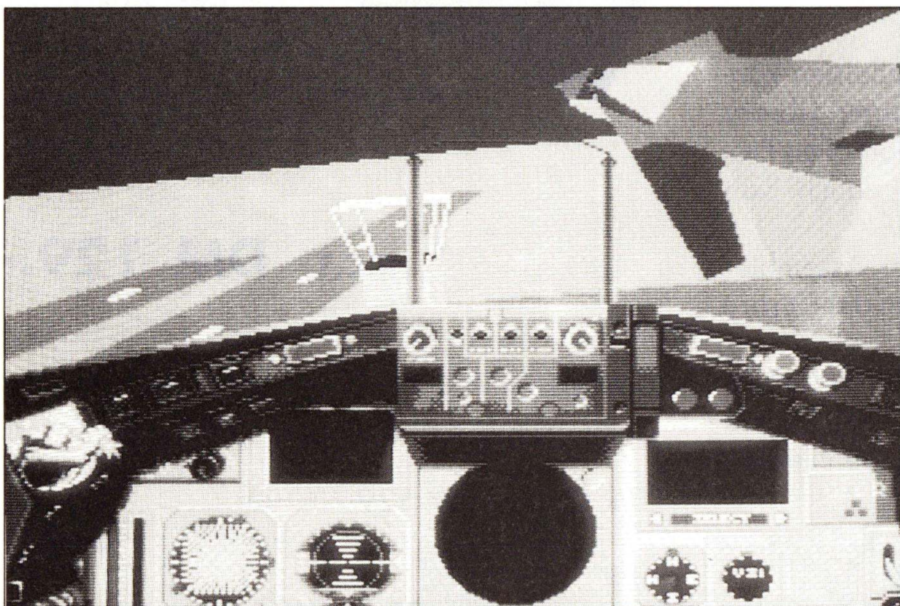
Um die wenigen Plätze an der Sonne raufen sich aber auch zur Zeit eine ganze Menge Produkte. So hat ACCOLADE nach längerem Schweigen nicht nur beschlossen, in London ein europäisches Büro zu eröffnen, sondern auch THE DUEL (Test Drive II) und zwei Scenario-Disks dazu herausgebracht. Die bereits legendäre Sportwagenssimulation bietet jetzt neue Kurse und Autos. Eine ebenso lange Wartezeit hatte AIRBALL, ein Geschicklichkeitsspiel um einen Luftballon in einem Labyrinth, der zwischendurch immer wieder aufgepumpt werden möchte. Microdeal hat sich damit sicher ebenfalls einen der besseren Plätze gesichert. Ob allerdings der Ball bei KICK OFF von Anco auch abhebt oder besser getreten werden sollte, muß sich noch zeigen. Fußballspiele werden bei uns in der Redaktion bekanntlich besonders genau unter die Lupe genommen. Etwas leichter hat es da UBI-Soft mit Skateball. Das "Eisgekicke" des französischen Softwaregiganten ist bereits auf dem ATARI ST erschienen und machte dort einen ganz passablen Eindruck. Die Graphik wird bei der AMIGA-Version identisch sein.

Gute Graphik, bereits in der zur Zeit vorliegenden PC-Version, verspricht auch das letzte Ballspiel unter den Neuvorstellungen - JACK NICKLAUS GOLF. Nun war ja gerade der Brite Nick Faldo etwas besser bei seinem Sieg im Masters, aber mit diesem Programm hat ACOLADE ganz sicher ein ASS gelandet. Das absolute ASS unter den Programmen der ECTS war allerdings POPULOUS - vertrieben von ELECTRONIC ARTS, geschrieben von BULLFROG. Dieses Spiel hat alle Chancen, zum bisher besten Spiel für den AMIGA gekürt zu werden. Eine ausführlichere Besprechung finden Sie an anderer Stelle im Heft (Tip für Freaks: bin bereits in Level 272 BADQAZ..., damit Sie nicht zu sehr nachhinken, das Codewort von 207 ist ALPTMET).




*Auch deutsche Firmen waren vertreten, im Bild sehen Sie die Vorstellung von TOM & JERRY von MAGIC BYTES.*

Aus der Rolle fällt auch ACTIVISION - erstens mit einer großen Zahl von Neuveröffentlichungen:
Timescanner, Bomber, SDI, Millenium 2.2 und anderen (more to follow next issue) und dann auch noch mit dem Rollenspiel PROPHECY. Prophecy ist eine Mischung aus Action- und Rollenspielelementen mit mehr als 60 verschiedenen Waffen und sieben Dungeon-Levels. Eine ähnliche Mischung bietet auch wieder CINEMAWARE mit LORDS OF THE RISING SUN. Ort der Handlung ist Japan in der Zeit der Shogune, und wie bei Bob Jacobs Truppe zu erwarten, ist die Graphik wieder brillant. Gute Ggraphik wurde auch bei zwei anderen Neuerscheinungen präsentiert: BLOOD MONEY von Psygnosis und ALIEN LEGION, das unter dem Label PEGASUS erschienen ist, und von GAINSTAR in England vertrieben wird. ALIEN LEGION ist ein Spiel im Stil von Impossible Mission - mit Graphik in Comicstil - und wurde von uns bereits an anderer Stelle vorgestellt.



*Bomber ist ein neues Spiel von Activision.*

Erfreulicherweise waren bei der ECTS auch deutsche Firmen vertreten - Magic Bytes präsentierte TOM UND JERRY - ein Hüpf- und Springspiel, CACHET war mit CRASH & BURN und QUASAR vertreten, und RAINBOW ARTS stellte GRAND MONSTER SLAM vor. Eine gute Nachricht für Fans von SIERRA habe ich mir bis zum Schluß aufgehoben - die Jungs, die uns Leisuresuit Larry geschenkt haben, sind soeben mit SPACE QUEST III fertig geworden. ■

VON MARTIN SILBERNAGL

# FULL SPEED

## Zweihundertsiebenundzwanzigtausend Assembler im Test oder: Wie Sie die Erde retten könnten

Die Raumschiffe kamen immer näher. Durch ihre neue Taktik hatten die Adobes es geschafft, unsere halbe Flotte auszulöschen. An den Gesichtern der Mannschaft war abzulesen, daß sie alles aufgegeben hatten. Doch Commander Mark Kandis dachte gar nicht daran, aufzugeben: "Unser Gefechtscomputer muß umprogrammiert werden! Schnell, wir haben fast keine Zeit mehr." Mark war eben der schwache Punkt in der Angriffstaktik aufgefallen, und er wußte plötzlich, wie er seine Heimat Terra retten würde. Er hetzte zur Konsole und aktivierte den Eingabemodus.

Ohne Probleme änderte er das alte Programm ab und brachte den Computer dazu, das neue zu übersetzen. Das Busy-Feld leuchtete auf - doch nur wenige Sekunden später ging im Raumschiff das Licht aus. Nur auf dem Computerbildschirm war noch etwas zu sehen:

Guru Meditation #00000004.FFFFFFE2

Mark Kandis war verzweifelt - was sollte er tun? Der Gefechtscomputer war ausgefallen und der große Angriff stand kurz bevor. Verzweiflung stand in seinem Gesicht. Niemals dürften die Adobes in Besitz des neuartigen Gefechtscomputers kommen. Mark setzte sich auf seine alte BMW, die er benutzte um die langen Strecken im Raumschiff zu überwinden, betätigte den KICKSTARTER, und raste mit dem Rechner unterm Arm zur äußeren Schleuse, wo eine kleine Raumphäre auf ihn wartete.



Déjà Vu? Ist Ihnen so (oder wenigstens so ähnlich) Ihr Assembler auch schon mal abgestürzt? Ja, Sie mein' ich. Waaas, er ist ihnen noch nie abgestürtzt? Sie Lüger (kein Fehler, sondern Dialekt)... Ach - Sie haben gar keinen. Ja dann. Wissen Sie denn überhaupt, was ein Assembler ist? Net? Na ja, is' ja nich' so schlimm.
Ich erklär's Ihnen gleich. Aber zuerst wollen Sie sicher wissen, wie die Geschichte ausgeht: Also... unsere Helden von der Erden treiben steuerlos im Weltenraum und sehen die Flotte der Adobes (was auch immer das sein mag) auf sich zukommen. Während Commander Kandis den Gefechtscomputer rettet, greift sich der erste Offizier Markus Mohr das Funkgerät und erzählt den Aliens, daß die Chancen, daß exterrestrisches Leben existiert, 1:10 99 stehen. Intelligent wie die Adobes sind, sehen die das ein und lösen sich in hellblaue und rosa Wölkchen auf. Schöne Geschichte, oder?

## Zum Unwesentlichen

Wer ist ein Assembler? Wenn Sie in der Schule Englisch haben/hatten, kennen Sie vielleicht das Wort "to assemble". Aber ich schätze, das hilft Ihnen auch nicht viel weiter.

Was also tut der/die/das Assembler (bei diesen Computer-Anglismen weiß man nie so genau, welches Geschlecht die haben)? Um das zu veranschaulichen muß, ich etwas weiter (ca. 2km) ausholen: In ihrem Kleinstrechner sitzt (im unerweiterten Zustand) eine Zentraleinheit mit dem Namen MC68000 (wer möchte schon so heißen), die nur relativ wenige Anweisungen kennt. Um das Leben am Computer zu erleichtern, wird diese Zentraleinheit meist nicht direkt programmiert, sondern mit höheren Programmiersprachen. Sehr wahrscheinlich kennen Sie BASIC (Wenn nicht, stellen Sie die Zeitung wieder ins Regal - die Kickstart ist keine Motorradzeitschrift). Hier holt sich der Interpreter während der Ausführung Programmzeile für Programmzeile und überstetzt sie in Maschinensprache, also in die Anweisungen, die der MC68000 kennt. So ähnlich auch bei den Compilersprachen, nur werden hier die Programme vorher ganz übersetzt und laufen dadurch schneller.
Was war aber in der Computersteinzeit, als es noch keine Hochsprachen gab? Man programmierte in LoLeLas (Low Level Languages). Anders gesagt: Man programmierte die CPU (Zentraleinheit) direkt in der jeweiligen Maschinensprache.

## Weshalb?

Wenn Sie vielleicht etwas mitdenken, könnte Sie nun interessieren, warum Sie denn dann überhaupt noch in Maschinensprache programmieren sollen, wo es doch so viele Hochsprachen gibt.

## Deshalb!

Einmal sind die Interpretersprachen zu langsam und die Compiler erzeugen (auf dem AMIGA) viel zu schlechten Code. Zudem sind viele Dinge (Nachladen von Disk im Interrupt...) in den Hochsprachen (inklusive C) nicht oder nur schwer machbar.
Allerdings muß man immer die Relation im Auge behalten: Manche Sachen sind in einer Hochsprache zehnmal schneller gemacht. Ich persönlich schreibe meine Programme immer teils in C für die komplexen Sachen, teils in Maschinensprache für die zeitkritischen, hardwarenahen Sektionen.

## Zum Thema

Da aber der MC68000 nur Zahlen als Befehle interpretiert und Menschen sich Zahlen relativ schlecht merken können, hat man für jede Zahl ein Kürzel (ein Mnemonic oder einen Opcode) eingeführt. Dieses Kürzel wird dann durch ein Anhängsel und die Vorzeichen der Daten, die dem Mnemonic folgen, genau spezifiziert. Natürlich kann ich nicht erwarten, daß Sie aus dieser Erklärung schlau geworden sind - deshalb ein klitzekleines Beipiel im lustigen Schaubild Nummer eins (gegen Ende). Wenn Sie sich daran sattgesehen haben, stellen Sie sich vielleicht die Frage: Wie mache ich aus dem ASCII-Text (so schön er ja aussieht) ein ausführbares Programm?

## Beim Thema

Ja, damit wäre ich endlich beim Thema. Ein Assembler ist ein mehr oder weniger langes und komplexes Programm, das solche Texte in ausführbaren Code übersetzen kann. Aber damit sind dessen Fähigkeiten noch lange nicht ausgeschöpft, denn sonst wäre dieser Artikel doch ein wenig zu viel Aufwand, oder nicht? Es gibt also verschiedene Testkriterien:

## Geschwindigkeit

Geschwindigkeit ist bekanntlich keine Hexerei. Deshalb frage ich mich, warum so viele Assembler so langsam sind. Scheinbar wurden hier oftmals Warteschleifen implementiert, um den Anwender zu ärgern - denn irgendwo muß ein Programmierer ja seinen Frust loswerden.

## Lieferumfang

Wichtigster Punkt hier ist das Handbuch (als Schutz gegen Raubkopierer), das in allen Fällen recht dick ist. Allein der Informationsdurchsatz läßt manchmal zu wünschen übrig. Doch auch die Software, die auf Disk mitgeliefert wird, geht hier in die Bewertung mit ein - Beispielprogramme sind gerade für Anfänger besser als jede weitschweifige Erklärung im Handbuch.

## AMA-Kompatibilität

Eigentlich sollte jeder Assembler vom Prinzip her dasselbe machen, aber manche verfügen über die Grundaufgabe (Mnemonics zu übersetzen) hinaus noch über andere Fähigkeiten: Die meisten können zum Beispiel Macros erzeugen (mehrere Befehle zu einem neuen Pseudo-Mnemonic zusammenfassen). Diese Zusatzfähigkeiten werden normalerweise in den Sourcecode (als sogenannte Pseudo-Opcodes) mit eingebaut.
Um aber Sourcecode (ASCII-Quelltexte) von einem Assembler auf den anderen zu übertragen, sollten alle über denselben "Wortschatz" verfügen. Eine gute Vergleichsgrundlage bildet hier der AmigaDOS Macro Assembler (AMA) von Metacomco, den ich seinerzeit (vor etwa drei Jahren) zusammen mit dem AmigaDOS Developer's Manual gekauft hab'. Mit ihm sind (weil er der einzige war) so ziemlich alle Assemblersachen des Betriebssystems gemacht worden. Auch die Sources auf

den alten Fish-Disks oder in den Entwicklungshandbüchern sind mit ihm entwickelt worden.

## Funktionen

Zudem müssen die anderen Assembler aber noch zusätzliche Funktionen zum AMA besitzen, denn sonst würde sie ja niemand kaufen. Zu diesen Funktionen gehört z.B. das Erzeugen von ortsfestem Code oder die Möglichkeit, Binärdateien einzubinden.

## Optimierung

Wenn Sie ein Programm geschrieben haben, muß das noch lange nicht optimal sein - Ihre kleinen "Fehler" sollte der Assembler auf Wunsch wegassemblieren. Jedoch ist die ganze Geschichte mit der Optimierung ein zweischneidiges Schwert: Es kann passieren, daß dadurch Bugs ins Programm geraten, nach denen man sich totsucht (am schlimmsten ist das Ganze, wenn man selbstmodifizierenden Code benutzt).

## Komfort

Da existieren also Assembler, mit denen man praktisch nicht arbeiten kann, ohne nach spätestens zehn Minuten den Drang zu verspüren, den heimischen Gummibaum hochzuklettern, weil zum Beispiel der eingebaute Editor viel zu langsam ist. Zudem gibt es welche, bei denen man den eigenen Quellcode in Null Komma Josef gelöscht hat.

# Der AmigaDOS Macro Assembler von Metacomco

Dieser Assembler existierte bereits 1985 in der Version 10.178 und ist heute in der Version 11.0 zu haben, bei der sich praktisch nichts geändert hat. Metacomco hat damals einen soliden Assembler herausgebracht, dessen Vorteile bestimmt nicht in der

### Geschwindigkeit

liegen. Das mag daran liegen, daß der AMA selbst in einer Hochsprache (in C) geschrieben worden ist. Wer gerne lange Programmodule benutzt, dem ist von diesem Programm nur abzuraten. Jedoch kann man mit diesem Teil ganz gut arbeiten, wenn man alles schön auf viele Module verteilt und nacher mit A-, B- oder C-Link alles zusammenlinkt.

### Lieferumfang

Zum AMA wird das AmigaDOS Developer's Manual mitgeliefert, das neben einigen anderen Kapiteln auch eines über die Benutzung des AMA beinhaltet. Nicht zuwenig und nicht zuviel ist auf den 23 großen Seiten zu finden - als Nachschlagewerk ideal, für Anfänger weniger, zumal es in englisch geschrieben ist.
Natürlich befinden sich auf der Disk die Includes, mit denen die Programmierung des Betriebssystems sehr einfach möglich ist, und auch einige spärliche Beipiele, die einem nur wenig bringen. Daß der AMA aus dem Entwicklungspaket stammt, merkt man an den mitgelieferten CLI-Befehlen: Außer dem Assembler "ASSEM" gibt es Funktionen, um Programme von Cross-Development-Maschinen zu laden.

### AMA-Kompatibilität

Ohne Zweifel darf man davon ausgehen, daß der AMA zu sich selbst kompatibel ist. Deshalb ist jedes weitere Wort in dieser Rubrik Verschwendung. Als alter Verschwender möchte ich dann genau hier meinen linken Turnschuh grüßen.

### Funktionen

Natürlich hat der AMA auch keine Funktionen mehr als der AMA. Deshalb entfällt auch diese Rubrik. Leider.

### Optimierung

Weder hat der AMA eine Optimierung, noch kann man diese abschalten. Somit wäre auch dieser Punkt geklärt.

### Komfort

Von Komfort kann bei diesem Teil keine Rede sein. Nicht nur, daß es langsam ist, nein - es verlangt auch vom User mehr als die anderen Assembler. So hat man z.B. mit dem Workspace ständig Ärger.

### Gesamteindruck

Mir kommt der AMA wie ein Relikt aus der AMIGA-Steinzeit vor. Als er damals im Entwicklungspacket lag, war er sehr nützlich. Doch jetzt gibt es zum Glück bessere. Da er allerdings den Standard darstellt, ist er manchmal ganz nützlich. Trotzdem würde ich gerade Anfängern vom Kauf abraten.

**Amiga Macro Assembler**

+ setzte den Standard
- langsam
- veraltet

# Der ASM von Lattice

Der ASM wird zusätzlich zum Lattice C geliefert, zum Alltagsbetrieb braucht man ihn allerdings nur, wenn man C und Assembler mischen will. Leider habe ich bis jetzt die Version 5.0 von Lattice noch nicht, so daß sich alle Aussagen auf die alte Version 4.0 beziehen. Ich schätze aber, daß sich kaum etwas geändert hat.

### Geschwindigkeit...

...ist auch bei diesem Assembler nicht das positiv hervorstechende Merkmal. Zudem muß man auch noch (wenn man weder Festplatte noch große RAM-Disk sein eigen nennen kann) die Ladezeit des Assemblers zur Assemblierungszeit dazurechnen, da der ASM genauso wie der AMA ein CLI-Befehl ist. Allerdings haben die CLI-Befehle gegenüber den Assemblern mit eingebautem Editor den Vorteil, daß man sie von einem Editor oder Makefile aus fernsteuern kann.

### Lieferumfang

Der Assembler ist im Handbuch ausreichend beschrieben - eigentlich kein Problem, da es nicht viel zu beschreiben gibt - und die Assemblerincludes bekommt man auch noch, aber sonst sieht es hier eher bös' aus.

## AMA-Kompatibilität und Funktionen

Auch hier ist's nicht gerade zum Besten bestellt. Der ASM kommt einem wie ein abgespeckter AMA vor.

## Optimierung

Der ASM kann Assemblertexte etwa genausogut optimieren wie meine Katze. Nochmals in klarerer Form: Der ASM hat keine Optimierung.

## Komfort

Der Komfort des ASM läßt sich am ehesten mit den Holzstühlen in unserer Schule vergleichen. Und die die sind verdammt unbequem.

## Gesamteindruck

Wie schon der große Eric R. sagte: "Einmal mit dem ist genug!"; der ASM kann nicht weiterempfohlen werden. Allein die Tatsache, langsamer zu sein als der AMA, hat etwas Unerhörtes an sich. Also Finger weg von diesem Teil, wobei ich wohlgemerkt nur vom Assembler spreche, nicht von Lattice C.

**ASM Lattice**

- \+ wird umsonst mitgeliefert
- \- langsamster Kandidat
- \- noch schlechter als AMA

# *Der AS von Manx (Aztec C)*

Der AS wird von Manx zum Lattice C ausgeliefert. Hier wird vom C-Compiler Assemblersourcecode generiert, den dann der AS erst zum Objektcode macht. Gegenüber Lattice C hat dieses Konzept den Vorteil, daß der Code von Hand nachgebessert werden kann. Wer sich Aztec-Code schon mal angeschaut hat, meint wahrscheinlich, daß das sehr nötig ist. Aber bei Manx war man so schlau, den Optimizer nicht in den Compiler, sondern in den Assembler einzubauen, so daß man mit dem AS einen sehr leistungsfähigen Assembler in der Hand hat.

## Geschwindigkeit

In puncto Geschwindigkeit ist der AS zwar auch nicht der beste, aber man kann auf erträgliche Weise damit arbeiten.

## Lieferumfang

Das Handbuch des Aztec C informiert ausführlich über den Gebrauch des Assemblers und über das Mischen von Assembler und C. Wie bei allen Assemblern, kann man hier der Dokumentation eine gute Note ausstellen. Offensichtlich der Grund für die guten Handbücher: Keiner der Assembler war kopiergeschützt.
Wie der AMA und der ASM auch, ist der AS ein CLI-Befehl und externe Editoren wie der KickED (Werbung muß sein) sind gefragt. Denn der mitgelieferte Editor Z mag auf UNIX-Rechnern mit 20 Mips ja ganz gut sein, aber auf dem AMIGA...

## AMA-Kompatibilität

Die alten Fish-Disk-Demos ließen sich ohne Anstand assemblieren, auch die Macrodefinitionen sind auf den Punkt übernommen worden. Hier braucht man sich keine Sorgen zu machen, wenn man nach den Reference Manuals programmieren will.

## Funktionen

Da der Assembler praktisch ein Teil des Compilers ist, sind natürlich einige spezielle Funktionen dazugekommen. FAR/NEAR, CODE/DATA sind zwei von diesen. Der Assembler beherrscht die Möglichkeit, allen Code und alle Variablen automatisch relativ zu A4 zu adressieren. Das kann man auch ausnutzen, wenn man in purem Assembler arbeitet. Weiterhin erkennt der Assembler leere Registerlisten, die per REG-Anweisung definiert worden sind, und läßt das movem dann weg. Im großen und ganzen also nichts Aufsehenerregendes bei den Zusatzfunktionen, bis auf den Pseudoopcode MACHINE, der den Prozessortyp festlegt (z.B. 68020) oder die Unterstützung des Fließkommacoprozessors MC68881.

## Optimierung

Wie vorhin schon angesprochen, ist der Optimizer des Compilers im Assembler untergebracht. Neben den Standardoptimierungen (Quicks, Shorts...) wird ganz schön am Code rumgebastelt, um die Dellen des Compilers auszubügeln. Wem das alles zu unheimlich ist, der kann den Optimizer auch ausschalten (mit -n). Das ist relativ wichtig, denn manchmal flippt der Assembler bei handgestricktem Code einfach aus. Auch hat mir der große Eric R. aus seinem persönlichen Nähkästchen erzählt, daß AS Mist baut, wenn man movem benutzt, um Daten durch den Speicher zu husten.

## Komfort

Bei diesem Teil läßt sich so ziemlich alles ein- bzw. ausschalten. Außerdem wird durch das mitgeliferte "make" die angeborene Faulheit aller Programmierer bestens unterstützt. Der Komfort ist für ein professionelles Arbeiten vollkommen ausreichend.

## Gesamteindruck

Wer mal so was echt toll Komplexes auf seinem AMIGA schreiben will, der sollte sich Aztec C kaufen und den Assembler für die zeitkritischen Teile benutzen. Für den Assembler alleine lohnt sich das Geld wirklich nicht. Auch um mal schnell 'nen Vorspann zu schreiben, ist der Assembler ziemlich ungeeignet, da er doch eigentlich auf andere Zwecke zugeschnitten ist. Trotzdem scheint es einige zu geben, die da anderer Ansicht sind (die AMIGA-Version von Drol wurde mit AS geschrieben), inzwischen jedoch belehrt wurden.

**AS Manx**

- \+ wird umsonst mitgeliefert
- \+ sehr gute Optimierung
- \+ unterstützt Assember/C-Mischung sehr gut
- \- 'Optimizer' spinnt manchmal, da nur für Compiler gebaut

-->

# Der CAPE 68K von Inovatronics

Beim CAPE handelt es sich wieder um ein reines Assemblerpaket, nach Angaben des Herstellers zur schnellen Entwicklung schneller Programme. Wer das nachprüfen will, kann sich eine voll funktionsfähige Vorversion auf PD besorgen.

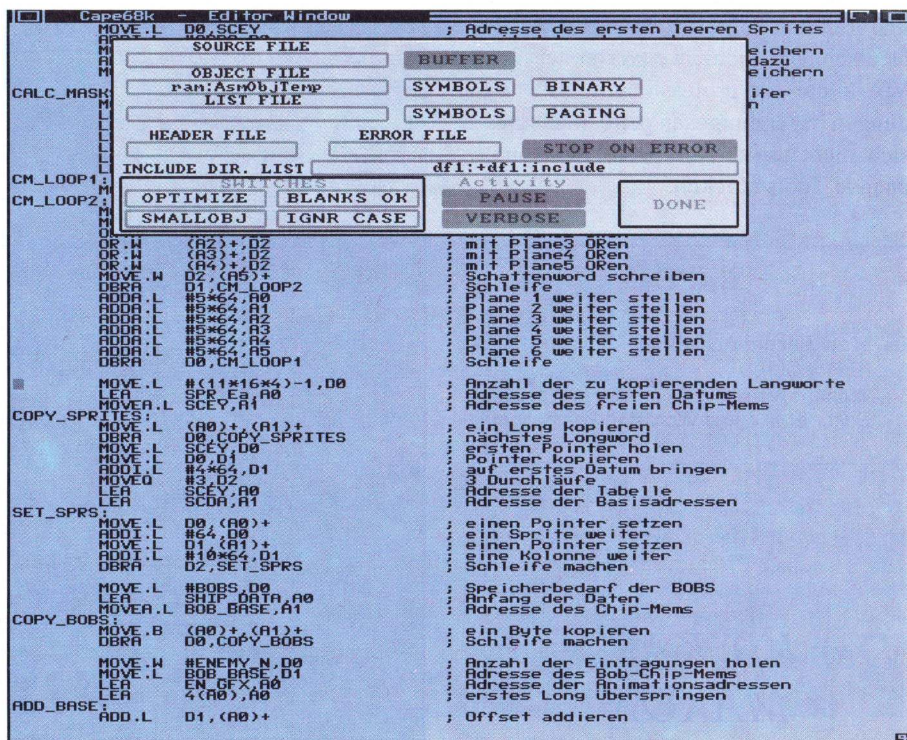
## Geschwindigkeit

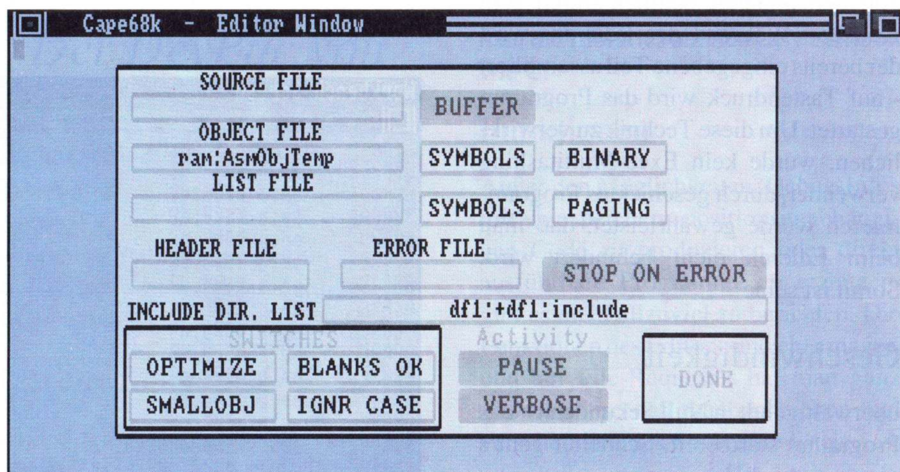
Seine Geschwindigkeit läßt (wohl wegen den vielen Pseudoopcodes) leider etwas zu wünschen übrig. Auch beim CAPE muß man ab etwa tausend Zeilen Quellcode anfangen, in verschiedene Module zu zerteilen, um dann den Linker zu bemühen.

## Lieferumfang

Der Lieferumfang ist für AMIGA-Verhältnisse als bombastisch zu bezeichnen. Den Assembler bekommt man zweimal (!): einmal in einen EMACS/Wordstar-kompatiblen Editor integriert und einmal als CLI-Befehl. Auch Beispielprogramme sind zwei vorhanden. Einziger Schatten: Die Includes fehlen, obwohl der Assembler in der Lage ist, die "genormten" Includes von Como zu benutzen.

Die Anleitung ist wohl eher für Leute gemacht, die schon ein bißchen Bescheid wissen, ansonsten ist sie aber recht brauchbar.




## AMA-Kompatibilität

Die AMA-Kompatibilität kann dem Cape wohl voll zugeschrieben werden. Allein ein paar Pseudoopcodes existieren nicht unter dem gewohnten Namen.

## Funktionen

Wenn man nach der Anzahl der Pseudoopcodes urteilt, müßte der Cape mit unzähligen Zusatzfunktionen ausgestattet sein. Da jedoch mehr als ein Drittel der Pseudoopcodes mit der gleichen Funktion belegt ist wie ein anderer, ist die Fülle nicht ganz so groß, wie man erwartet. Dennoch sind einige (seeehr) nützliche Dinge dabei: Mittels der BASEREG-Option kann man Ähnliches erreichen wie bei Aztec mit der NEAR-Option. Auch erwähnenswert ist der auf Wunsch erzeugte DEBUG-Hunk, der sowohl für Anfänger als auch für Profis manchmal unentbehrlich ist.

## Optimierung

Die Optimierung ist schlicht und einfach als grandios zu bezeichnen. Mittels TRASHREG kann man ein Datenregister angeben, das der Optimizer dann verwendet, um den Code schneller zu machen. Die optimierten Zeilen werden beim Auslisten mit einem Ausrufezeichen gekennzeichnet, aber wem selbst das zu sehr in seine Intimsphäre greift, der kann den Optimizer ausschalten.

## Komfort

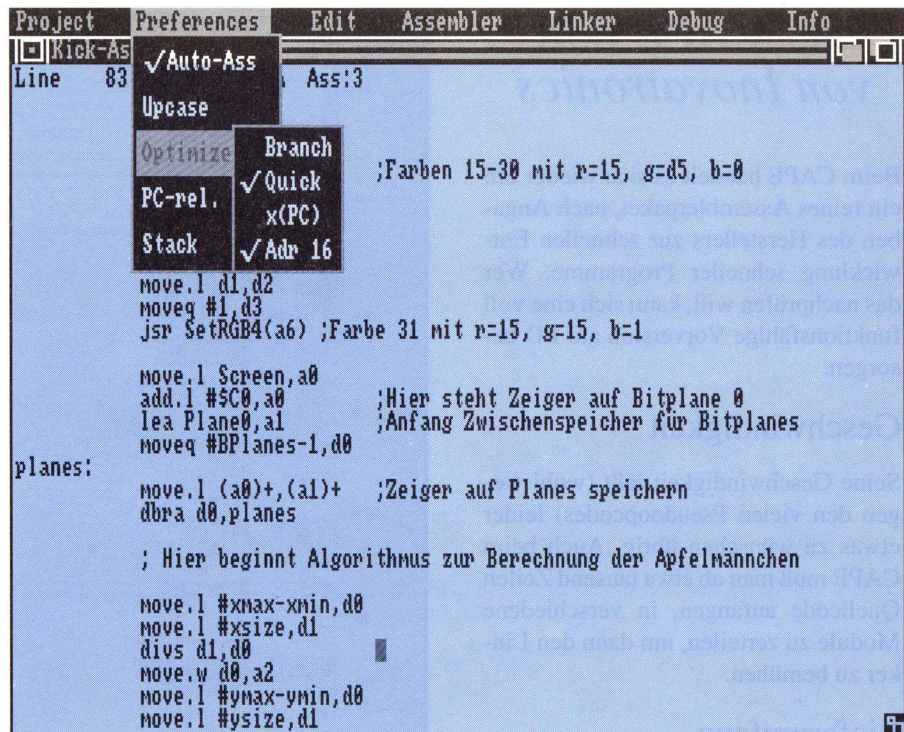
Durch die zwei Versionen wird eine größtmögliche Flexibilität gewahrt - wem der Texteditor nicht gut genug ist, der kann dann mit dem CLI-Befehl arbeiten. Wer lieber mit Menüleisten spielt, kann im Editor programmieren. Zudem ist der Assembler ARexx-kompatibel, er kann also mit anderen ARexx-Programmen (z.B. CygnusED) kommunizieren.

## Gesamteindruck

Beim Cape handelt es sich um einen Assembler, den man sich auf alle Fälle

mal reintun sollte, auch wenn er in Deutschland nicht weitverbreitet ist. Vor allem für professionelle Anwendungen hat er einiges in petto, man muß sich nicht umständlich selbst entsprechende Tools stricken.

> **CAPE**
>
> + in zweifacher Ausführung enthalten (Editor/CLI)
> + versteht ARexx
> + Editor EMAX und Wordstart kompatibel




## *Der KICKASS von MAXON*

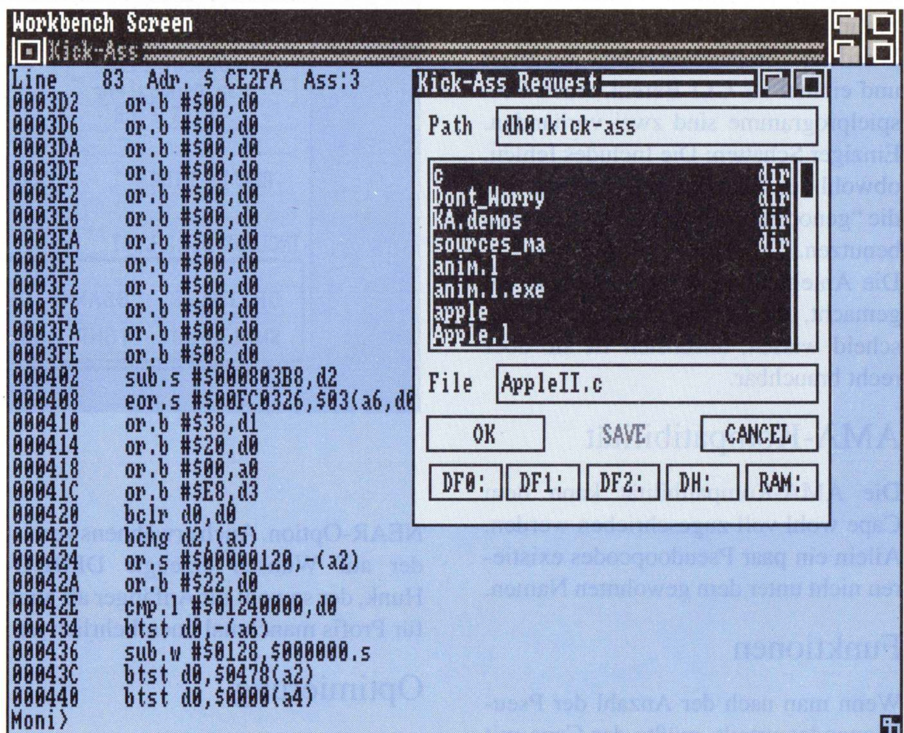
Dieser Assembler unterscheidet sich von den anderen deutlich durch sein Konzept: Während des Edierens wird der bereits eingegebene Teil assembliert - auf Tastendruck wird das Programm gestartet. Um diese Technik zu verwirklichen, wurde kein Exec-Multitasking verwendet, durch geschicktes Programmieren wurde gewährleistet, daß man beim Edieren nicht behindert wird. Somit ist seine

### Geschwindigkeit

überwältigend: in Null Sekunden ist das Programm startbereit. Schneller geht's nimmer. Auch der

### Lieferumfang

ist nicht ohne. Zum ausführlichen deutschen Handbuch bekommt man einen Assembler, bei dem in nur 60 Kilobyte ein Editor, eine Kommandooberfläche mit vielen nützlichen Befehlen, ein interner Linker und ein Debugger integriert sind. Natürlich sind auch Beispiele auf der Disk. Die




### AMA-Kompatibilität

ist dafür leider nicht so toll. Der KICK ASS ist für Betriebssystemprogrammierung deshalb weniger, aber nicht ungeeignet, wie die mitgelieferten Beispiele zeigen. Seine Stärke liegt eher bei den zusätzlichen

### Funktionen

Die sind nämlich enorm. Angefangen von den einfachen Menübefehlen, um zum Beispiel externe Module so einzubinden, daß das Programm auch mit diesen auf Tastendruck gestartet werden kann, über die Möglichkeit, den Sourcecode (sehr) gepackt abzuspeichern und zu laden, bis hin zu den Befehlen der Kommandooberfläche, von denen einer zum Beispiel automatisch ein GLOBL-

File für einen bestimmten Bereich des Quellcodes erzeugen kann, geht die Palette (ich bin wieder in Hochform: pro Absatz ein Satz).

## Optimierung

Wenn man sich den KICK ASS genauer anschaut, bemerkt man, daß er jede Zeile, anstatt sie, wie alle anderen Assembler, direkt in den Quelltext zu übernehmen, interpretiert. Ist sie syntaktisch oder logisch falsch, wird darauf mit einer Fehlermeldung reagiert. War sie dagegen richtig, wird geprüft, ob es sich um eine Abkürzung handelt. Gibt man also "ml #500,d0" ein, wird daraus automatisch die Zeile "move.l #500,d0". Ist die Zeile soweit interpretiert, setzt der Optimizter ein, und sofern optimiert werden kann, wird der Code umgewandelt. Dann erst wird die Zeile in den Sourcecode übernommen und angezeigt. Die Probleme mit den Optimizern entfallen also, weil man immer darüber informiert ist, wie der Code denn nun im Endeffekt aussieht. Wer Optimizer aber trotzdem nicht mag, kann die verschiedenen Optimierungen einzeln an- und ausschalten.

## Komfort

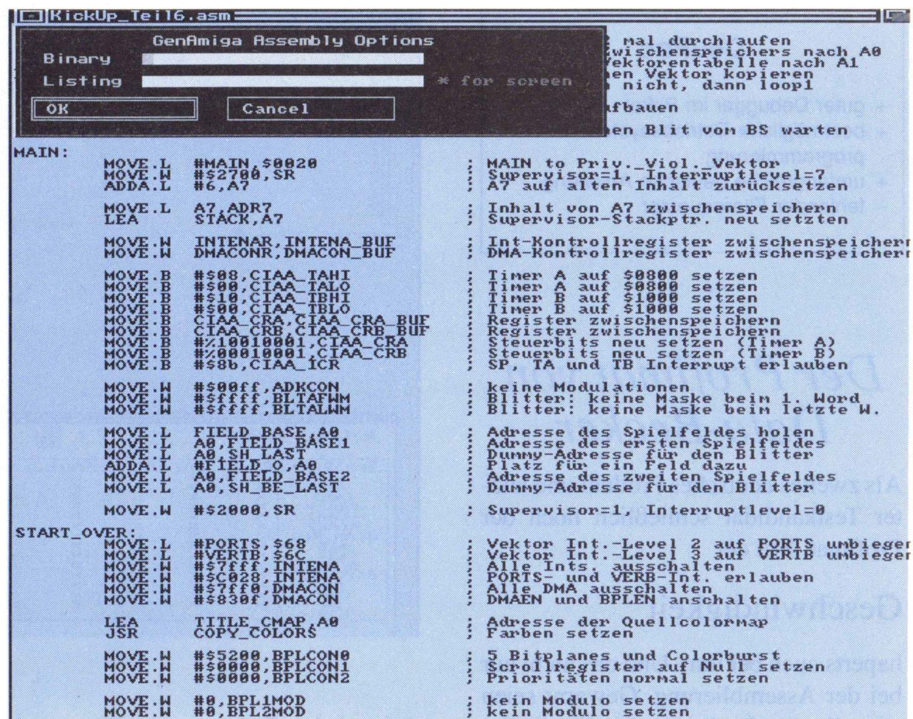
Etwas Komfortableres als den KICK ASS gibt es eigentlich nicht, gerade für Anfänger (Syntaxcheck und leichte Bedienung) und für den gestressten Profi (Zeitersparniss durch Direktassemblierung) ist KICK ASS ideal.

## Gesamteindruck

Fürs pure Assemblerprogrammieren ist der KICK ASS wunderbar geeignet. Durch sein einmaliges Konzept entlastet er den Programmierer vor allem im Dauerbetrieb sehr - auch Anfänger profitieren davon.

### KICKASS

+ Komplettpaket in nur 60KB
+ komfortable Oberfläche
+ Programme immer startbereit
+ optimale Kontrolle über Optimizer
+ wahlweise internes oder externes Linken
+ sofortige Syntaxprüfung
+ deutsche Anleitung, niedriger Preis
- schlechte AMA-Kompatiblität
- keine Makros




# *Der DevPac von HiSoft*

Auch ein reines Assemblerpaket ist der DevPac, der seit neuestem in der Version 2.0 zu haben ist. Da diese bis Redaktionsschluß nicht eintraf, wurde V 1.2 zum Test herangezogen.
In puncto

## Geschwindigkeit

liegt der DevPac ganz vorn. Selbst der KICK ASS kann nicht mithalten, wenn er ein ganzes Programm neu assemblieren muß.

## Lieferumfang

Der Assembler kommt auf einer bootfähigen Diskette, auf der neben dem Assembler-/Editor-Programm noch ein Programm zur Voreinstellung der Assemblerparameter und ein sehr guter Debugger sind.
Das Handbuch der deutschen Version ist vom Umfang her wohl als optimal zu bezeichnen, als Nachschlagewerk aber fast schon etwas zu dick geraten. Die Includes sind vorhanden und können voll ausgenutzt werden. Die

## AMA Kompatibilität

ist voll gegeben. Mit den Includes kann man die Libraries ähnlich easy aufrufen wie C.

## Funktionen

Außer den Möglichkeiten, Debug-Infos mitabzuspeichern, positionsunabhängigen Code zu produzieren oder direkt ausführbare Programme zu erzeugen, gibt es nicht allzuviel zu berichten. Die Funktionen des Editors sind eher mager, und auf File-Requester hat man ganz verzichtet. so daß man sich schon mal versehentlich das Sourcefile löscht.

## Optimierung

Sämtlichen Problemen sind die Programmierer aus dem Weg gegangen, indem sie genialerweise keine Optimierung eingebaut haben.

## Komfort

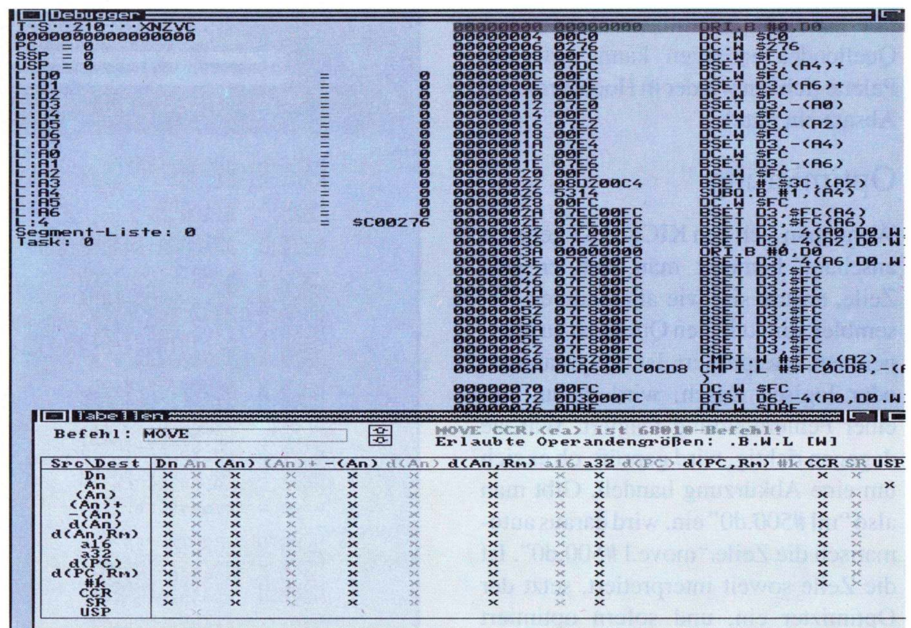
Wie ich weiter oben schon erwähnt habe, fehlen die file-requester, ansonsten ist das Teil von der Bedienerfreundlichkeit her o.k.

## Gesamteindruck

Wie ich noch weiter oben schon erwähnt habe, ist das Teil ein Speed-King und war lange Zeit mein Lieblingsassembler. Man kann gut mit ihm arbeiten, wenn man sich erst an ihn gewöhnt hat.

**DevPac**

+ guter Debugger im Paket
+ bestmögliche Betriebssystem-programmierung
+ umfangreiche deutsche Anleitung
- fehlender Filerequester




# *Der Profimat von Data-Becker*

Als zweites deutsches Produkt und letzter Testkandidat schließlich noch der Profimat. Mit der

## Geschwindigkeit

haperts auch bei ihm. Und das nicht nur bei der Assemblierung. Gewarnt seien alle, die den Profimat schneller bedienen, als er reagieren kann. Ein Guru ist nämlich die Folge solchen Handelns.

## Lieferumfang

Das Handbuch vom Profimat ist der absolute Hit: Alles ist so ausführlich erklärt, daß sogar mein Kühlschrank auf Anhieb alles verstanden hat. Das Hanbuch ist sogar ein richtiges Buch, kein Ringhefter oder ähnliches. Da es noch dazu in deutsch verfaßt ist, ist das wohl die ideale Anleitung für Anfänger.
Auf der bootfähigen Diskette befinden sich neben dem Assembler zahllose Hilfsdateien-Includes, die allerdings von den Standartincludes abweichen, da die

## AMA-Kompatibilität

sehr bescheiden ausgefallen ist. Am schlimmsten ist, daß man keine Objektmodule erzeugen kann, was eine professionelle Verwendung des Assemblers verhindert.
Dafür sind die anderen

## Funktionen

um so oppulenter ausgefallen: in vier Fenstern geht es rund. Der Editor, der Debugger, der Assembler und ein Hilfsfenster, in dem man sich z.B. alle Befehle samt Adressierungsart zeigen lassen kann, müssen auf dem Bildschirm untergebracht werden. Allerdings kann der Debugger insofern nicht überzeugen, als daß zu jedem Debuggen das File sowohl assembliert als auch eingeladen werden muß. Auch ist eine

## Optimierung

nicht vorhanden, was gerade für Anfänger, die ja nicht alle Taktzyklen im Kopf haben, von Vorteil wäre. Der

## Komfort

ist durch die reichhaltigen Menüleisten und Requester garantiert. Leider ist das teilweise auch der Grund für die Behäbigkeit des Teils.

## Gesamtbild

Ideal für die ersten Schritte und für ein kleines Programm nebenher, aber ungeeignet für den wirklichen "off road"-Einsatz, so lautet mein Urteil. Aber auch hier gibt es Leute, die mit dem IBYTE-Befehl PC-relative Routinen einbauen und so lange Programme geschrieben haben, ohne ganz durchzudrehen.

**Profimat**

+ beste Anleitung
+ Hilfestellungen im Programm
+- deutsche Bedienungsoberfläche
- keine Linkmodule
- nicht ganz ausgereiftes Konzept

# *Epilog*

Die Auswahl ist groß, auch wenn nicht alle Assembler in diesem Vergleich zur Sprache kamen, die Entscheidung fällt dadurch natürlich nicht leichter. Wenn Sie sich noch nicht sicher sind, welchen Assembler Sie sich anschaffen sollen, dann vertiefen Sie sich nochmal in diesen Bericht und versuchen nach den Kriterien auszuwählen die Ihnen am wichtigsten erscheinen.

```
; Beispiel eines Maschienenspracheprogramms.
; Ein Strichpunkt bedeutet immer,
; daß der Rest der Zeile überlesen wird.

        MOVE.L  #$AFFE,$4

; ^
; ein Opcode(Mnemonic) darf nie am Anfange einer Zeile stehen
;        ^
;       ein Datum soll bewegt werden
;             ^
;            mit der Größe Langwort (4 Byte)
;                ^
;                und dem hexadezimalen Wert 0000affe
;                         ^
;                        an die Adresse 00000004

        RTS

; Rückkehr an den Aufrufenden (z.B. CLI/DOS)
```

*Lustiges Schaubild Nummer Eins ist ein kleines Assemblerprogramm*

## Funktionsübersicht der Kandidaten

| | AMA | ASM | AS | CAPE | KICKASS | DevPac | Profimat |
|---|---|---|---|---|---|---|---|
| Linkmodule | Ja | Ja | Ja | Ja | Ja | Ja | Nein |
| 68010 | Nein | Nein | Ja | Ja | Nein | Nein | Ja |
| 68020/68881 | Nein | Nein | Ja | Nein | Nein | Nein | Nein |
| Optimierung | Nein | Nein | SehrGut | SehrGut | SehrGut | Nein | Ja |
| Fast/Chip | Nein | Ja | Ja | Ja | Nein | Ja | Ja |
| Makros | Ja | Ja | Ja | Ja | Nein | Ja | Ja |
| CLI-Befehl/Standalone | CLI | CLI | CLI | Beides | StAl | StAl | StAl |
| Direktassemblierung | Nein | Nein | Nein | Nein | Ja | Nein | Nein |
| Sourcekomprimierung | Nein | Nein | Nein | Nein | Ja | Nein | Nein |

*Schaubild 2: Die Fähigkeiten der Assembler im Vergleich. Hierbei zeigt sich, daß mache Assembler für gewisse Dinge besser sind als andere. Es scheint (noch) keinen ultimativen Assembler zu geben.*

## Zeitvergleich

| AMA | ASM | AS | CAPE68K | KICKASS | DevPac | Profimat |
|---|---|---|---|---|---|---|
| 2:31 | 2:48 | 1:50 | 2:08 | 1:19/0:0* | 0:47 | 2:23 |

* Bei eingeladenen Source-Code, also während der Programmentwicklung, benötigt KICK ASS keinerlei Assemblierungszeit und ist damit uneinholbarer Sieger dieses Wettbewerbs. Die erste Zeit ist die Assemblierzeit einer nicht im Editor befindlichen Quelldatei.

*Schaubild 3: Assemblierungszeiten in Sekunden. Die Assemblierung erfolgte in einer RAM-Disk; der LIST-Modus war ausgeschaltet. Bei den CLI-Assemblern wurde die Uhr gestartet, als die Titelzeile erschien.*

VON ROGER SCHMIDT

# PageStream

## Ein DTP-Programm setzt (Drucker-) Maßstäbe

*Desktop-Publishing heißt das Zauberwort, von dem seit langem jeder Computerbesitzer redet. AMIGA-User mit Nadeldruckern führten auf diesem Gebiet bislang ein Schattendasein. Damit ist jetzt Schluß. Ein neuer Stern am DTP-Himmel zeigt den Matrixdruckern, wo es lang geht.*

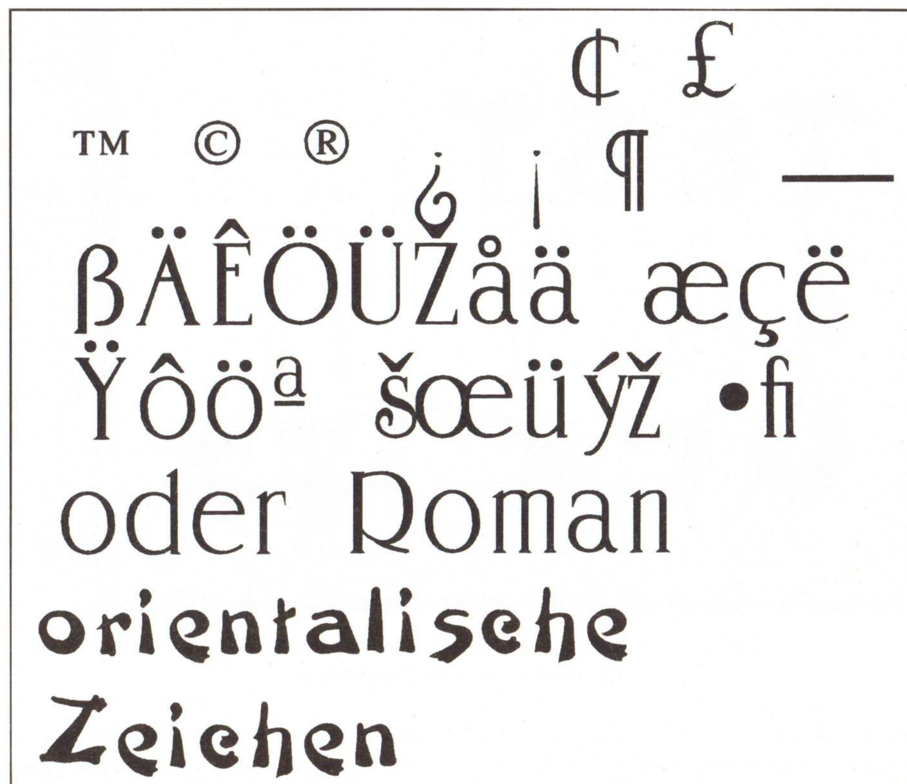
KICKSTART
KICKSTART
KICKSTART
KICKSTART
KICKSTART
KICKSTART
KICKSTART
KICKSTART

*Der Schrifttyp Thymes in achtfacher Variation*

Mit optimierten Zeichensätzen und neuen Druckertreibern wird Drucken ohne Laser-Printer zum reinsten Vergnügen. Der Name des neuen AMIGA-Sternenkindes: PAGESTREAM. Natürlich werden die Laserdrucker nicht außer acht gelassen, selbst postscriptfähige Drucker können angesteuert werden.

Jeder AMIGA-Besitzer wird wohl einmal in dieser Situation gewesen sein: Ein Freund kommt zu Besuch (selbstverständlich totaler Computerlaie), und Sie zeigen ihm voller Stolz Ihre Computeranlage nebst erst kürzlich erworbenem Matrixdrucker. Da kommt auch schon die quälende Frage. "Kann der Kasten auch Einladungskarten drukken?". Nach kurzer Denkpause und Überhören des Wortes "Kasten" bejaht man dies. Nachdem Sie Ihr geliebtes Stück per Netzschalter zum Leben erweckt haben und mit PageSetter oder

einem anderem DTP-Programm einen "Probeausdruck" zuwege brachten, kommt auch schon die nächste Bemerkung des ungeliebten Besuchers. "Die großen Buchstaben sind ja total eckig!". Nachdem Sie sich an diesem Punkt in eine Rechtfertigung über Bildschirmauflösungen und maximale DPI-Werte des Druckers verstrickt haben, zieht der Besucher meist noch mit einer Bemerkung wie "teures Spielzeug" ab. Man sitzt dann noch selbst mit starrem Blick minutenlang vor dem Ausdruck, betrachtet das Treppchen-Syndrom und schwört sich, nie wieder solchen Computer-Banausen seine Anlage vorzuführen. Mit dem Auftauchen der Workbench V 1.3 schöpften zwar einige AMIGA-Besitzer wieder Hoffnung, aber die Option "Smoothing" im Preference-Menue erzeugte aus einer großen Treppe auch nur zwei kleine Treppchen mit gerundeten Kanten. "Wenigstens haben sich die Druckzeiten auf ein erträgliches Maß reduziert", dachte so mancher, und blickte neidvoll auf Besitzer von Laserdruckern (neudeutsch: Laser-Belichter), die mit ihren zentnerschweren Ungetümen wahre Meisterleistung aufs Papier brachten. Doch jetzt gibt es auch für die guten alten Nadeldrucker wieder Hoffnung. Wie Phoenix aus der Asche, kam Ende April das Programm PageStream auf den Markt. Als der Autor dieses Artikels das frisch erworbene Programm endlich in Händen hielt und noch mißtrauisch die Schlagworte wie "DTP-Programm der zweiten Generation" und "optimierte Zeichensätze für Matrixdrucker" las, konnte noch niemand ahnen, daß dies den Beginn einer neuen Druckerära bedeutete. Aber da Tonerstaub für Laserdrucker seit eh und je krebserregend ist, und man seinem Nadeldrucker noch eine Chance geben wollte, ging es sofort ans Werk.




*Sonderzeichen oder orientalische Schrifttypen stellen für PageStream kein Problem dar.*

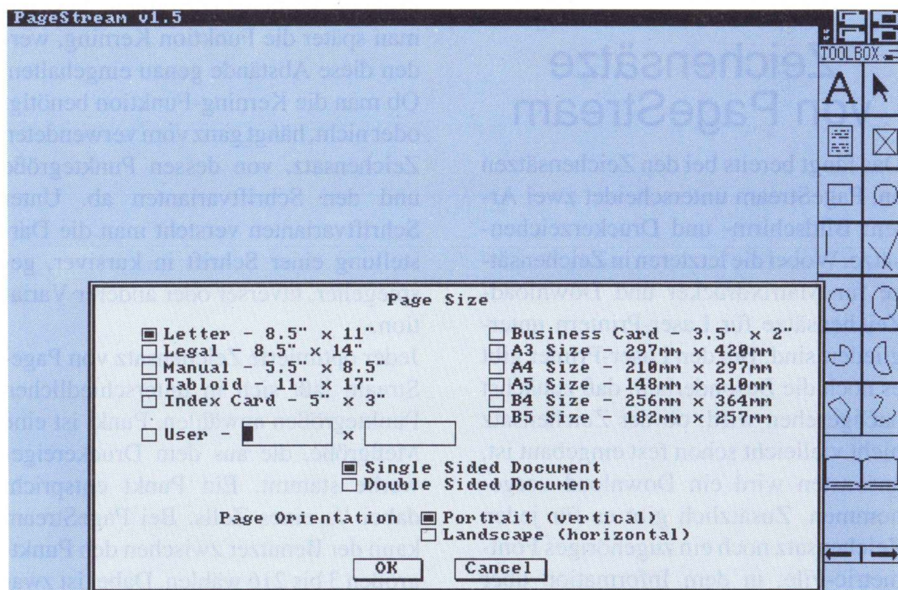



*Bild 1: Eine große Auswahl von verschiedenen Blattgrößen bietet das Auswahlmenü*

## Der Lieferumfang

Im Lieferumfang befanden sich zwei Disketten mit der Versionsnummer 1.5 sowie ein Handbuch im stabilen Ringordner. Nach kurzem Anlesen der Anleitung, leider noch auf Englisch, erfuhr man, daß die Grundausstattung, ein AMIGA mit 512 KB RAM sowie ein Laufwerk, ausreicht, um kleinere Dokumente zu erstellen (wohlgemerkt kleinere Dokumente). Wie sich später herausstellte, hat man aber bei dieser Ausstattung des öfteren mit Guru-Fehlermeldungen zu kämpfen. Als ideal kann man eine Ausstattung mit 1 MB RAM und 2 Laufwerken bezeichnen, wobei zusätzlicher RAM-Speicher besonders bei Dokumenten mit zahlreichen Grafikbildern anzuraten ist.

Als großer Pluspunkt des Programms kann der Umstand gewertet werden, daß PageStream nicht kopiergeschützt ist. Der Hersteller empfiehlt sogar das Anlegen von Sicherheitskopien oder, falls vorhanden, die Installation auf Festplatte. Dies zeigt wieder einmal, daß für ein gutes Anwenderprogramm der beste Schutz eine detailierte Anleitung ist. Denn trotz der gut aufgebauten Pull-

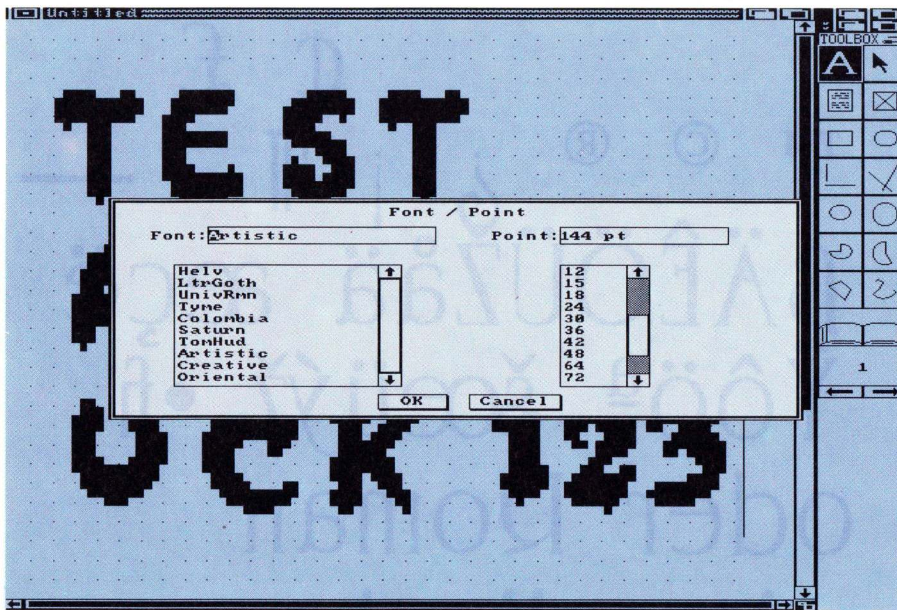



*Bild 2: Zeichensätze en masse und in allen Größen bietet die Font-Disk von PageStream.*

down-Menüs und Filerequester muß man die Anleitung immer wieder zu Rate ziehen. Dies liegt aber hauptsächlich an der Funktionsvielfalt des gesamten Programmpaketes.

## Zeichensätze von PageStream

Das fängt bereits bei den Zeichensätzen an. PageStream unterscheidet zwei Arten: Bildschirm- und Druckerzeichensätze. Wobei die letzteren in Zeichensätze für Matrixdrucker und Download-Zeichensätze für Laser-Printern untergliedert sind. Bei den Laser-Printer gibt es noch die Besonderheit, daß zunächst nachgesehen wird, ob der Zeichensatz nicht vielleicht schon fest eingebaut ist, ansonsten wird ein Download vorgenommen. Zusätzlich gibt es für jeden Zeichensatz noch ein zugehöriges Fontmetric-File, in dem Information über Abstand und Breite der einzelnen Buchstaben gespeichert ist.

PageStream unterstützt sowohl auf dem Bildschirm als auch beim Ausdruck den gesamten ASCII-Zeichenvorrat. Über die Kombination Control-C bzw. Control-D lassen sich alle länderspezifischen Sonderzeichen ansteuern, angefangen vom "German double s", $^1/_4$, $^1/_8$ usw. bis zu eingetragenes Warenzeichen. Ebenfalls vorhanden sind Spezialzeichen aus Frankreich, Norwegen, Schweden usw. Je nach Aussehen des Zeichensatzes kann es jedoch passieren, daß zwischen verschieden Buchstaben oder Spezialzeichen unerwünscht große Abstände entstehen. Deshalb können Sie bei PageStream für ganze Zeichensätze oder ausgewählte Buchstabengruppen Abstände vorwählen. Aktiviert man später die Funktion Kerning, werden diese Abstände genau eingehalten. Ob man die Kerning-Funktion benötigt oder nicht, hängt ganz vom verwendeten Zeichensatz, von dessen Punktegröße und den Schriftvarianten ab. Unter Schriftvarianten versteht man die Darstellung einer Schrift in kursiver, gespiegelter, inverser oder anderer Variation.

Jeder optimierte Zeichensatz von PageStream läßt sich in unterschiedlichen Punktegrößen anwählen. Punkt ist eine Meßgröße, die aus dem Druckereigewerbe stammt. Ein Punkt entspricht dabei $^1/_{72}$ eines Zolls. Bei PageStream kann der Benutzer zwischen den Punktgrößen 3 bis 216 wählen. Dabei ist zwar nicht jede Punktgröße in dieser Skala vorhanden, aber die Abstände zwischen den einzelnen Größen sind relativ kurz, so daß sich eigentlich immer eine passende Größe findet. Im Lieferumfang sind insgesamt 10 verschiedene Zeichensätze für Matrixdrucker enthalten. Wenn man dann diese druckeroptimierten Zeichensätze mit den unterschiedlichen Schriftarten und noch dazu in verschiedenen Größen darstellt, läßt es sich schon ganz ordentlich arbeiten. Wer noch mehr Vielfalt benötigt, kann noch insgesamt weitere fünf Font-Disketten erwerben, von denen jede jeweils drei weitere Zeichensätze enthält. Für Laserdruckerbesitzer werden Disketten mit Postscript-Bildschirm-Fonts und diversen Download-Fonts angeboten.

## Druckvielfalt

Bevor man mit der eigentlichen Arbeit, dem Entwurf einer Zeitungsseite, anfängt, will PageStream zunächst wissen, welches Blattformat beim späteren Ausdruck zur Verwendung kommt. Neben den üblichen amerikanischen Inch-Formaten wurde auch an die Europäer gedacht. Von DIN A5 bis DIN A3 sowie B4 und B5 läßt sich alles auswählen. Daneben gibt es noch ein Spezialformat zum Ausdruck von Visitenkärtchen. Danach wird der Benutzer noch gefragt, ob er in Portrait, das heißt vertikal bzw. in Landscape horizontal ausdrucken möchte. Unterstützt wird jeder Drucker, den man unter den gleich mitgelieferten Preferences V1.3 findet. Das Programm entscheidet dabei auch zwischen den DIN A4 und DIN A3-Modellen des einzelnen Herstellers. Besitzt man zum Beispiel ein NEC Pinwriter P6 Plus und wählt ein Blattformat von DIN A3, wird die komplette Seite auf 4 Blätter mit Überlappungen gedruckt.

Bei dem Auswahlpunkt Druckgröße ist normalerweise der Grundwert 100 Prozent angegeben. Er läßt sich bei Matrixdruckern bis auf 10000 Prozent, bei Postscript-Lasern sogar bis auf 65000 Prozent einstellen. Gleichzeitig sollte man jedoch dann die Funktion Tiling aktivieren, wodurch der Ausdruck auf vier oder gar neun Blätter verteilt wird. Die Funktion Thumbnails hingegen generiert Miniversionen der einzelnen Dokumente, so daß man innerhalb kürzester Zeit einen Überblick des Gesamtdokumentes erhält. Auch an PageStream-Besitzer mit Farbdrucker wurde gedacht. So gibt es verschiedene Funktionen zum Farbausdruck bis hin zum 4-Farben-Separationsverfahren, das reprofertige Vorlagen liefert. Die Funktion Transparenz druckt ein Dokument in seiner gespiegelten Form. Es bietet sich dann die Möglichkeit, dieses Dokument auf einen Overheadprojektor zu legen und auf einer darübergelegten Folie zu schreiben, ohne das Original zu verändern.

Die Option Write White druckt hingegen das Bild als Negativ aus. Zusätzlich

# DAS GFA-BASIC

# 3.0 BUCH

**Lieferbar!**

**KLAUS SCHNEIDER, OLIVER STEINMEIER, PETER FRITZEN**

Als optimale Ergänzung zum Handbuch des neuen GFA-BASIC 3.0-Interpreters bietet sich dieses Buch an. In zwei Hauptteilen wird zunächst eine systematische Einführung in die Programmierung von BASIC unter Berücksichtigung der besonderen Fähigkeiten von GFA-BASIC auf dem Amiga gegeben. Hier werden dem Neuling vom ersten Einzeiler bis zu abstrakten Datentypen alle Möglichkeiten der strukturierten Programmierung mit zahlreichen, durch Flußdiagramme transparenter gemachten Beispielen nahegebracht. Doch auch BASIC-erfahrene Programmierer lernen hier die neuen Strukturen kennen, die sich doch sehr von denen anderer Dialekte unterscheiden.

Der zweite Teil bauf auf dem ersten auf und vermittelt weitere Kenntnisse der Programmierung, anhand von Programmen, die wiederum ausführlich beschrieben und erklärt sind. Hier seien ein leistungsfähiges Grafikprogramm sowie zahlreiche Beispiele zur Betriebssystemprogrammierung genannt.




Die Benutzung der verschiedenen Libraries und Intuition-Funktionen wird detailliert erklärt, so daß die Verwendung dieser Möglichkeiten in eigenen Programmen keine Schwierigkeiten bereitet.

Durch zahlreiche Anhänge – neben vielen Tabellen finden Sie auch ein sehr ausführliches Stichwortverzeichnis – wird das Buch optimal ergänzt und kann problemlos zum Nachschlagen von Details benutzt werden.

## AUS DEM INHALT:

**Erklärung der Schleifen- und Programmstrukturen**
- FOR-NEXT, WHILE-WEND, REPEAT-UNTIL, DO-LOOP
- Prozeduren, Funktionsunterprogramme und Verzweigungen
- Rekursion
- Beispielprogramme

**Variablentypen und Arrays**
- numerische und Zeichenkettenvariablen
- Arrays zur Aufnahme großer Datenmengen

**Multitasking in GFA-BASIC**
- Reagieren auf Ereignisse
- Zeitabhängige Prozeduraufrufe

**Programmentwicklung**
- Programmplanung und -entwurf
- strukturierte Programmierung
- TOP-DOWN-Prinzip
- Fehlersuche
- Debugging-Möglichkeiten

**Dateiverwaltung**
- sequentielle Dateien
- Random-Access-Dateien
- Funktionen und Befehle zur Diskettenverwaltung

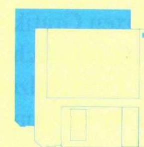
**Grafikprogrammierung**
- Die vielseitigen Möglichkeiten
- Turtlegrafik, Spriteprogrammierung
- Arbeiten mit mehreren Bildschirmen
- HAM-Modus
- Entwicklung eines vielseitigen Grafikprogramms

**Sound- und Sprachprogrammierung**
- Die Möglichkeiten der Befehle
- Verwendung der Anweisung in eigenen Programmen

**Abstrakte Datentypen**
- Verkettete Listen
- Bäume

**Betriebssystemprogrammierung**
- Pulldownmenü-Steuerung
- Requester-Einsatz in Programmen
- Intuition und Graphics-Library
- Aufruf von Systemroutinen

**Für Einsteiger ★ Fortgeschrittene ★ und Profis**



**ÜBER 500 Seiten EINSCHLIESSLICH PROGRAMMDISKETTE**

Dem Buch liegt eine Programm-Diskette bei mit über 150 Übungs- und Beispielprogrammen

BUCH & DISKETTE KOMPLETT
HARDCOVER **DM 59,-**

Preise sind unverbindlich empfohlene Verkaufspreise

**Heim Verlag**

*Heidelberger Landstr. 194*
*6100 Darmstadt-Eberstadt*
*Telefon 0 61 51-5 60 57*

Schweiz: DataTrade AG
8021 Zürich
Langstr. 94

**BESTELLCOUPON AMIGA GFA 3.0 BUCH**

Bitte senden Sie mir ______ St. DAS GFA-BASIC **3.0 BUCH** einschließlich Programm-Diskette für DM **59,–**
zzgl. DM 5,– Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ______________________________

Straße, Hausnr. ______________________ PLZ/Ort ______________________

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte.

kann man beim Ausdruck einer zusammengehörigen Folge von Seiten, deren Reihenfolge bestimmen. Entweder wird zunächst Seite eins, zwei, drei usw. gedruckt oder ganz einfach vom letzten Blatt an: ... drei, zwei, eins. Das hat den Vorteil, daß bei längeren Dokumenten umständliches Umsortieren entfällt. Auch die Density (Punkteauflösung) des jeweiligen Druckers läßt sich bequem vom Druckerrequester aus einstellen. Bei Matrixdruckern geht sie je nach Fähigkeiten des Druckers bis zu 360 * 360 DPI. Bei den höchsten Auflösungen steigt die Druckzeit zwar proportional an, jedoch lassen sich hier auch Maximalergebnisse mit hohem Schwärzungsgrad erzielen. Ebenso läßt sich die Anzahl der zu druckenden Kopien festlegen. Von eins bis 65000 ist jeder Wert zugelassen.

## Text und Grafik-Einbindung

Wer mehr als nur eine Schriftart verwenden möchte, muß sich auch Gedanken über die Gestaltung seines Dokuments machen. Überschriften, Grafikbilder, Skizzen, gefüllte Flächen und überlappende Text stellen höchste Anforderungen an Desktop-Publishing-Programme. Doch auch hier hat PageStream einiges zu bieten. Der Benutzer kann neben Textboxen Grafikbereiche, eigene Objekte mit unterschiedlichen Füllmustern und Freihandzeichnungen erstellen. So können zum Beispiel alle "Meisterwerke" mit einer eigenhändigen Unterschrift versehen oder eigene Strichzeichnung eingebunden werden, ohne jedesmal vorher ein extra Grafikprogramm zu benutzen. Mit der Importfunktion lassen sich sowohl Bitmap-orientierte Grafiken von DPaint und IFF-ILBM-Bilder einladen als auch Vektorgrafiken im Aegis Draw Plus-Format.

An Texten läßt sich alles verarbeiten, was im ASCII-Format vorliegt. Texte im Word Perfect-Format werden sogar 1:1 übernommen. Sobald man Text oder Grafiken in PageStream übernommen hat, erscheinen sie in einer neuen Box, von wo man sie dann mit Hilfe der Kopier- und Ausschneidefunktion an die vorgesehene Stelle des Dokuments setzt. Dort kann dann noch die Größe und der Platz verändert werden. Falls Texte nicht ganz in eine Box gehen, so muß der Benutzer die Box entweder vergrößern oder er verknüpft die zu klein gewordene mit einer neuen Box. Will der Benutzer in der Mitte eines Textblocks eine Grafik plazieren, muß er entscheiden, wie der Text an dem Grafikbild vorbeifließt. PageStream bietet hier fünf Möglichkeiten. Zunächst kein Textfluß, das heißt, zuerst kommt Text, dann die Grafik und dann wieder Text. Rechts und links vom Grafikbild bleibt der Platz frei. Oder man wählt die Option, daß der Text rechts oder links des Grafikbildes entlang fließt. Oder das Grafikbild ist von allen Seiten mit Text umgeben. Selbstverständlich läßt sich auch hier die Entfernung der Text- und Grafikboxen zueinander einstellen. Boxen können mit Hilfe der Rotate-Funktion auch schräg gestellt werden. Wer Preisschilder drucken möchte, lernt diese Option schnell zu schätzen. In Ein-Grad-Schritten läßt sich jede Box um die X-, Y- oder Z-Achse spiegeln. Mehrere Text- oder Grafikboxen können sich auch überschneiden, wobei der Benutzer wählen kann, in welcher Reihenfolge Text- und Grafikabschnitte hintereinander liegen. Zur Textbearbeitung in den einzelnen Boxen stehen alle Standardtextverarbeitungsbefehle zu Verfügung. Neben Einfügemodus, Text suchen, ersetzen und löschen gibt es noch ein PageStream-eigenes Korrekturprogramm, das, da nur mit englischen Vokabeln ausgestattet, bei der Bearbeitung deutscher Texte wenig hilfreich ist. Wer über genügend Geduld verfügt, kann jedoch seine eigene Wortschatzbibliothek aufbauen. Vielleicht findet auch ein Leser einen Weg, das Rechtschreiblexikon der deutschsprachigen Word Perfect-Version einzulesen. Bei der Textbearbeitung taucht auch immer wieder die Aufgabenstellung auf, mehrere Kommandos wie etwa Buchstabe löschen, eine Zeile tiefer, Buchstabe löschen usw. auszuführen. Man kann solche Befehlsfolgen als Macros auf die Funktionstasten F1 bis F10 legen und abspeichern. Beim nächsten Laden des Programms sind diese Grundeinstellungen dann automatisch vorhanden.

Genauso lassen sich die Pfade für das Laden und Abspeichern von Dokumenten, Text- und Grafikfiles und der Zeichensätze festlegen. Für die Skalierung der Dokumente kann der Benutzer zwischen neun gebräuchlichen Einheiten wählen. So gibt es neben Zoll, Picas oder Punkten auch Zentimeter und Millimeter.

## Seitendarstellung

Zur Ansicht des Dokumentes bieten sich gleichfalls verschiedene Möglichkeiten. So kann PageStream zum Beispiel zwei Seiten gleichzeitig anzeigen. Dabei ist es zwar möglich, daß der Text nicht mehr deutlich lesbar ist, aber man kann so überprüfen, ob die Gestaltung der Seiten optisch zueinander paßt. Es

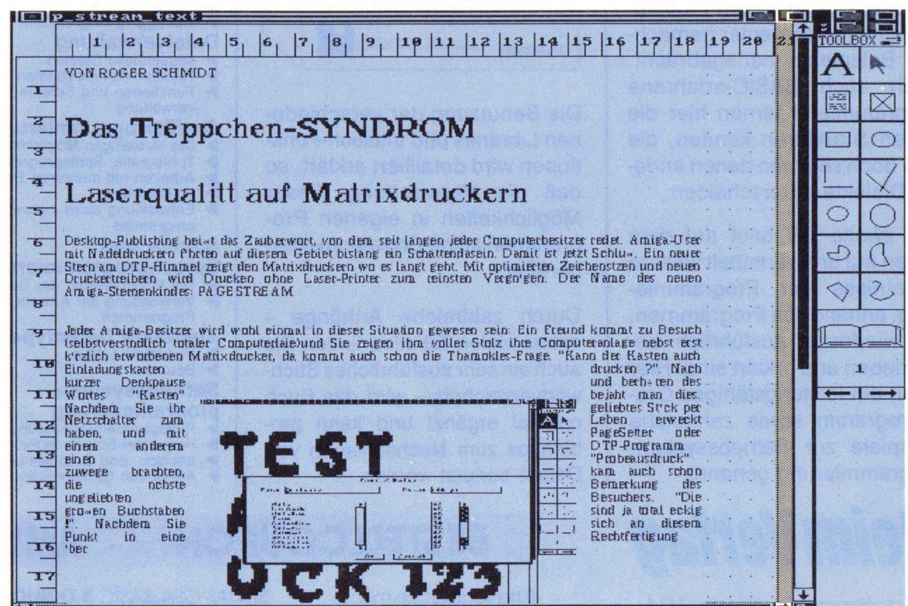



*Bild 3: Bei diesem Beipiellayout lassen wir den Text um ein Objekt (Bild) herumlaufen.*

kann sich hier auch um Seiten handeln, die nicht direkt hintereinander liegen. Auf die gleiche Art läßt sich so eine ganze Seite anzeigen. Zusätzlich ist noch die Option einer 50prozentigen Verkleinerung sowie einer 200prozentigen Vergrößerung des Dokumentes vorhanden. Mit der Zoom-Option kann man dann noch einen benutzerdefinierten Ausschnitt bis zu 1500 Prozent vergrößert darstellen. Verknüpfungen von Textboxen lassen sich im gleichen Menü anzeigen. Grafikbilder können während der Arbeit mit PageStream abgeschaltet werden, um die Arbeitsgeschwindigkeit zu erhöhen. Textboxen werden auf die gleiche Art inaktiviert. Beim nächsten Bildschirmaufbau erscheint dann anstatt des Textes eine schwarze Fläche. Der gesamte Bildschirmaufbau geht dann um ein Vielfaches schneller und erleichtert somit das Arbeiten. Bei Bedarf lassen sich diese Funktionen jederzeit verändern.

## Fazit

Insgesamt kann man das Desktop-Publishing-Programm PageStream als gelungen bezeichnen. Eine Vielzahl von Funktion, unterschiedliche Zeichensätze mit Sondersymbolen und gute Druckergebnisse sind die Pluspunkte dieses Programms. Fehlender Kopierschutz und Bedienerfreundlichkeit sowie ein gut gemachtes Handbuch zeigen neue Wege für professionelle Anwenderprogramme. Einzige Minuspunkte bilden der Interlace-Modus und die noch (?) englische Bedienerführung. Wenn man sich jedoch die Ausdrucke auf Nadeldruckern betrachtet, läßt sich sagen:

A new Star is born!

### PageStream

+ einfache Bedienung
+ hervorragende Druckergebnisse
+ optimierte Zeichensätze
+ sehr viele Funktionen
+ kein Kopierschutz

- englisches Handbuch
- Interlace-Modus

*Hersteller:*
*Soft Logik*

*Anbieter:*
*CSS*
*Auf der Warte 46*
*6367 Karben 1*
*Tel. 06039-5776*

Jetzt für AMIGA


CHIP · JUGEND+TECHNIK · ZDF COMPUTER CORNER

URKUNDE

GOLDENE DISKETTE 1987

für das beste Lernprogramm
Schirmherr:
Der Bundesminister für Forschung und Technologie
Dr. Heinz Riesenhuber

verliehen an

DIRK OWERFELDT

für

VOKABULA

programmiert auf
Atari ST

Dieses Siegerprogramm des Programmierwettbewerbes GOLDENE DISKETTE 87 ist die Grundlage für den neuen Vokabeltrainer aus dem Heim-Verlag:

# AMIGA-Learn

Vers. 3.5

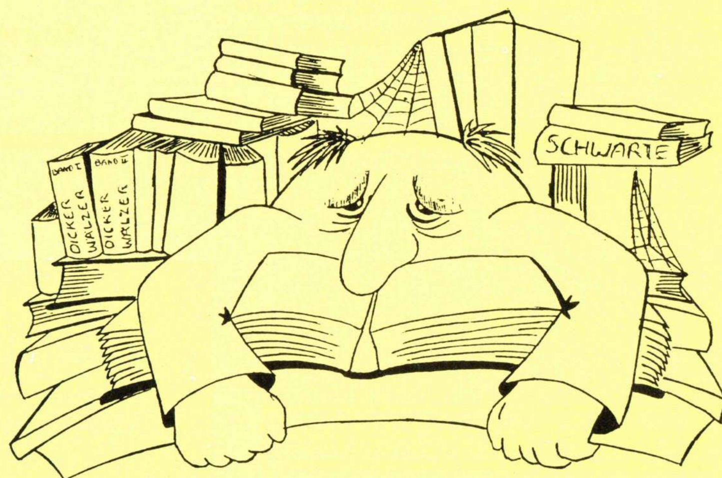
Das Urteil der Jury, **bevor** das Programm völlig überarbeitet und nochmals verbessert wurde:

„VOKABULA ist eines der wenigen Vokabelprogramme, das wirklich neue Eigenschaften für diese Kategorie von Lernprogrammen aufweist... Es wurde in C geschrieben und benutzt geschickt und effizient die grafische Benutzeroberfläche...
Dirk Owerfeldt hat mit VOKABULA das bisher professionellste Vokabel-Programm geschrieben, das beim Wettbewerb um die GOLDENE DISKETTE eingereicht wurde."

**Und das sind die Leistungsdaten:**

- bietet Spaß beim Lernen durch Grafik und Sound
- Unterstützt den vollen europäischen Zeichensatz (bequemer Zugriff durch die Maus
- Fehlerhäufigkeit der Vokabel wird berücksichtigt
- Mehrere Bedeutungen eines Wortes werden berücksichtigt
- „Intelligente" Auswertung der Benutzereingaben u. a. spezielle Berücksichtigung unregelmäßiger Verben (bei Eingabe von 'to go' erfolgt keine Fehlermeldung, sondern es werden die anderen Formen nachgefragt)
- Bei offensichtlicher Ähnlichkeit der Wörter wird wahlweise ein zweiter Versuch zugelassen
- Vielfältige Möglichkeiten des Lernens und der Abfrage (Deutsch-Fremdsprache, Fremdsprache-Deutsch, Multiple-Choice, lernen durch optische Rückkopplung)
- Sortieren der Vokabeln nach mehreren Parametern
- konsequente Realisierung verschiedener Lerntechniken:
  u. a. – Karteikarten-Konzept
  – Lernen in fester Reihenfolge
  – zufällige Stichproben
  – Abfragen, bis alle Vokabeln gekonnt werden
- Jederzeit Bewertung möglich, die den Lernerfolg anzeigt und mit einem Kommentar motiviert
- integriertes Lernspiel HANGMAN
- Wörterbuchfunktion, sucht Übersetzung für eine Vokabel
- Ausgabe aller Vokabeln bzw. aller falschen oder richtigen Vokabeln auf Bildschirm oder Drucker
- Trotz Einordnung der Vokabeln nach Lektionen oder Wissensgebieten ständig schneller Zugriff auf alle Vokabeln (nur durch Größe des Speichermediums begrenzt)

Im Lieferumfang befinden sich über 1600 englische Grundwortschatzvokabeln in zwei Schwierigkeitsstufen, sowie mehrere Dateien mit wichtigen Vokablen, etwa Wendungen und Struktur- oder Ordnungswörter.




**DM 69,–**

* alle Preise sind unverbindlich empfohlene Verkaufspreise

## BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle: ______St. AMIGA-Learn á 69,– DM
zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ______________________

Straße, Hausnr. ______________________

PLZ, Ort ______________________

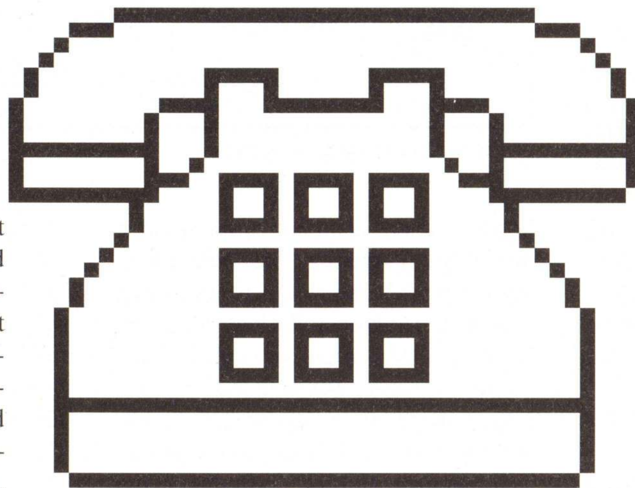
Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte.

**Heim Verlag**
*Heidelberger Landstraße 194*
*6100 Darmstadt-Eberstadt*
*Telefon 06151-56057*

SCHWEIZ
**DataTrade** AG
Langstr. 94
CH-8021 Zürich

VON REINER MOLL

# BTX
# WAS IST DAS?



Die Abkürzung BTX steht für Bildschirmtext und wird bundesweit zum Ortstarif angeboten. Im Dialog mit dem Bildschirm wird die Möglichkeit geboten, Bankgeschäfte, Versandbestellungen, Auskünfte und Serviceleistungen zahlreicher Anbieter in Anspruch zu nehmen. BTX ist 24 Stunden am Tage geöffnet. Seit wenigen Wochen sind auch Gäste willkommen, um das Angebot einmal kostenlos kennenzulernen.

Nun werden werden Sie sich fragen, wie man eine derartig tolle Sache preiswert nutzen kann, wo doch ein BTX-Terminal mehr kostet als ein AMIGA 2000? Das Problem löst sich auf eine elegante Weise, der AMIGA ist voll bildschirmtexttauglich und wird von der Post zugelassen. Commodore überlegt, in Kürze zu jedem neuen AMIGA ein BTX-Programm dazuzulegen, das eine besonders gute Grafik haben wird. Allerdings arbeitet der AMIGA mit völlig anderen Signalen, als sie zur Zeit im Telefonnetz üblich sind. Wie beide Systeme trotzdem sehr gut miteinander auskommen können, darüber möchte ich berichten. In der Schaltung des AMIGA können die Impulse nur zwei Zustände haben, Spannung oder keine Spannung, +5 Volt oder 0 Volt. Im Telefonnetz dagegen können nur Töne, wie sie z.B. aus dem Lautsprecher kommen, übertragen werden. Man spricht hier von digitaler oder analoger Betriebsweise. Schaltet man nun zwischen den AMIGA und das Telefon einen Wandler, der digitale in analoge Signale wandeln kann, steht der Teilnahme des AMIGA an BTX nichts mehr im Wege. Als Wandler sind sogenannte Modems oder Akustikkoppler (Koppler) gebräuchlich. Da ein Koppler einfacher angeschlossen werden kann als ein Modem, gehe ich auf diesen näher ein.

*Unter BTX versteht man ein grafikfähiges Mitteilungs- und Informationssystem, das über das Telefonnetz der Bundespost läuft. Mit einem BTX-Terminal kann man den Dienst von zahlreichen Anbietern in Anspruch nehmen.*

Damit der Koppler die Töne der BTX-Zentrale verstehen kann, muß er der V.23-Norm entsprechen. In dieser Vereinbarung ist festgelegt, daß der Koppler einen Datenstrom von 1200 Bit/s empfangen und mit 75 Bit/s senden muß. Zwischen dem AMIGA und dem Koppler gilt die V.24-Norm. Die V.24-Buchse des Kopplers muß mit der seriellen Buchse des AMIGA mit einem passenden Kabel verbunden werden. Dieses Kabel können Sie leicht selbst anfertigen.

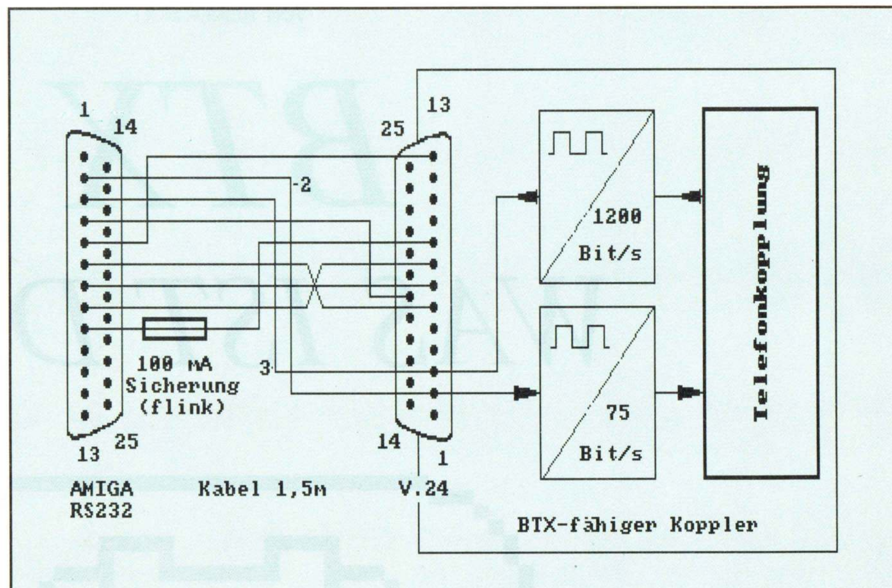
Auf der mit "2" bezeichneten Leitung sendet der AMIGA dem Koppler die Daten mit 75 Bit/s. Auf der Leitung "3" erhält der AMIGA die BTX-Daten mit 1200 Bit/s. In den BTX-Daten sind die BTX-Grafik und die Texte enthalten, die auf dem Monitor gezeigt werden. Auf den übrigen Leitungen verständigen sich die Geräte bei der Übertragungsprozedur.

Ist der Koppler mit dem Flachbandkabel am AMIGA angeschlossen, können Sie den Rechner einschalten und das BTX-

Programm starten. Vom Programm werden Sie dann aufgefordert, die BTX-Zentrale anzuwählen. Am Telefon wählen Sie die Nummer "190" und warten auf den BTX-Ton. Dann können Sie den Hörer in die Gummimuscheln des Kopplers legen. Achten Sie dabei auf das aufgedruckte Telefonsymbol, damit der Hörer nicht falsch herum eingelegt ist. Hat alles geklappt, begrüßt Sie das BTX-Titelbild. Diese bestätigen Sie mit <Return>, und die BTX-Zentrale begrüßt Sie als Gast in ihrem Netz.

Ganz nach Wunsch können Sie sich alle angebotenen Tafeln ansehen und das sind einige Tausend. Die Verbindung wählen Sie auf dem rechten Ziffernblock. Eingeleitet wird die Wahl stets mit einem Stern '*' und abgeschlossen mit Raute '#'. Dieses Zeichen liegt auf der 'Enter'-Taste. Wählen können Sie wie beim Telefon mit Ziffern, aber auch mit Namen. Wählen Sie z. B. '*commodore#', so begrüßt der AMIGA-Hersteller mit einem eigenen Auswahlmenü. In die BTX-Zentrale zurück gelangen Sie, besonders wenn Sie sich nicht mehr zurechfinden, einfach mit '*0#'.

Möchten Sie einmal viel Geld überweisen? Bei BTX ist das möglich, wählen Sie doch einmal die Telebanken an. Hier können Sie das Geld mit vollen Händen ausgeben. Vergessen Sie bitte nicht, dem Verfasser einen kleinen Geldbetrag zukommen zulassen. Weitere Möglichkeiten aus dem Angebot sind: Devisenumrechnungen, Börsenkurse, Kreditzinsen-, Rentenberechnungen, Reisebürobuchungen usw. Das elektronische Telefonbuch '*1188#' hilft Ihnen bei der Suche nach dem richtigen Anschluß. So können Sie z. B. ermitteln, wie oft Ihr Name in deutschen Telefonbüchern vorkommt. Als AMIGA-Besitzer haben Sie den Vorteil, daß Sie sich nichts zu notieren brauchen. Auf Wunsch speichert das BTX-Programm alle gezeigten Bildtafeln automatisch, wie mit einem Videorecorder. Nach Beendigung der BTX-Verbindung können Sie sich alle Seiten noch einmal in Ruhe ansehen, auf eine Diskette sichern und sogar auf einem Drucker ausdrucken. ■



*Schaltskizze des Anschlußkabels*

Von Frank Schäfer

# Lass' Batchdateien für Dich arbeiten

## Teil 1: Grundlagen und Praxistips

*Batchdateien, Startup-Sequencen, der EXECUTE-Befehl..., viele Begriffe und wenig brauchbare Informationen. Mit diesen Artikeln wollen wir viel Licht in den Dschungel des CLI strahlen lassen.*

Im Teil eins geht es um die Grundlagen zu Batchdateien, ein paar Praxistips, Anmerkungen zum CLI, um File-Flags, und was der Dinge mehr sind.

Teil zwei beschäftigt sich dann näher mit den Batchdateien, mit Sprüngen, Schleifen, Bedingungen, Rekursion, Batchdateien & Workbench sowie weiteren kleinen Tips & Tricks.

Teil drei soll sich mit ARexx (u.a. eine einfache programmierbare Möglichkeit, Daten zwischen Programmen auszutauschen) auseinandersetzen, denn CygnusEd, TxEd, TeX, WShell sind alle Programme, die ein ARexx-Interface besitzen. Wir wollen einmal zeigen, was man mit ARexx so machen kann. Nur sollte man sich mit dem CLI etwas auskennen, bevor man mit ARexx beginnt.

Bevor wir uns mit den Batchdateien beschäftigen, sollten Sie erst einmal den Kasten mit den Begriffserklärungen lesen. Wenn das geschehen ist, bitte hier weiter.

Nach dem Lesen der Begriffserklärungen sollten nun die in diesem Text verwendeten Fachausdrücke klar sein. Die Bedeutung der Batchdateien ist recht groß, denn sowie Sie eine Workbench-Diskette in den Rechner schieben, arbeitet der Rechner eine Batchdatei ab. Diese Datei ist unter dem Namen Startup-Sequence bekannt. Als erstes wollen wir uns einmal die Startup-Sequence Ihrer Workbench anschauen. Um sich dieses File anzusehen, kann man wie

### Begriffserklärungen

| | |
|---|---|
| *Batchdatei* | Eine Textdatei mit CLI-Befehlen, die nacheinander aufgerufen und abgearbeitet werden |
| *Booten* | Vorgang, der das Betriebssystem einliest (Kickstart, Workbench) |
| *Default(Wert)* | Grundeinstellung bzw. Grundwert, der benutzt wird, falls der Benutzer nichts anderes eingibt |
| *Device* | Ein Gerät wie ein Laufwerk oder eine Festplatte. Im Zusammenhang mit dem AMIGA ist damit allerdings häufig auch ein Programm gemeint, das wie ein (Hardware-) Gerät funktioniert, z.B. zu einer RAM-Disk gehört das Programm RAM-Handler. |
| *Device-Name* | Der Name, den ein Device trägt z.B. RAM: , DF0: , DF1: , DH0: , JH0: , CON: , PRT: , ... . Wie man sieht, ist das Kennzeichen eines Device-Namens der Doppelpunkt am Ende des Namens. |
| *Logische Devices* | Ein Bezeichner (Name), der einem Directory zugeordnet ist. Diese Zuordnung kann mit dem Befehl ASSIGN erfolgen. Einige Bezeichner haben eine Defaulteinstellung wie zum Beispiel S: , C:, DEVS: , L: , LIBS: , SYS:. Zu diesem Thema siehe den Artikel "Der ASSIGN-Befehl" in der KICKSTART 2/89. |
| *RAM-Disk* | Ein im Arbeitsspeicher simuliertes Laufwerk. Eine RAM-Disk ist das mit Abstand schnellste Speichermedium. Besonders wichtig für Leute mit nur einem Laufwerk. |
| *Resident* | Mit dem RESIDENT-Befehl in den Arbeitsspeicher kopierte Programme werden ähnlich wie in der RAM-Disk verwaltet. Falls Programme von der RAM-Disk gestartet werden existieren sie doppelt: einmal in der RAM-Disk und einmal im Arbeitsspeicher. Von RESIDENTEN-Programmen existiert immer nur eine Kopie, auch wenn sie mehrfach gestartet werden. Es lassen sich leider nur manche Programme resident machen (siehe Fileflags) |
| *Shell* | zu deutsch Muschel. Meint aber im Computerbereich eine Benutzerfreundlich gestaltete Eingabemöglichkeit für Kommandos. |
| *Semikolon* | Das Semikolon ";" wird im CLI verwendet, um Kommentare zu kennzeichnen, z.B.: -> DIR ; Hallo dies ist ein Kommentar! Alles nach dem Semikolon wird ignoriert. |

folgt vorgehen: auf der Workbench den Notepad starten, die Menüfunktion "Open" anwählen und als Datei "s:Startup-Sequence" eingeben.
Wenn man die Eingabe mit "okay" bestätigt, wird die Datei geladen. Nach kurzer Zeit ist die Startup-Sequence und damit die Befehlsfolge, die der Rechner beim Booten abarbeitet, sichtbar. Nun speichern Sie mit der Menüfunktion "Save As" die Startup-Sequence unter dem neuen Namen "s:Startup-Sequence.bak" ab. Jetzt basteln wir unsere eigene Startup-Sequence zusammen.

## 1,2,3... eine Startup-Sequence herbei

Mit der Menüfunktion "New" können Sie den Inhalt des Notepads löschen.
Bevor Sie beginnen, die neue Startup-Sequence 1 einzugeben, besprechen wir diese Batchdatei erst einmal. Die erste Zeile hat nur eine Aufgabe, sie schreibt den in Anführungszeichen stehenden Text in das CLI-Window. Mit "LoadWb" wird die Workbench (was auch sonst?) gestartet. In der dritten Zeile wird mit Hilfe des SETMAP-Befehls die deutsche Tastaturbelegung geladen (ja, genau die Äs und Ös). Die letzte Zeile startet das sich im Directory Utilities befindende Notepad-Programm als einen neuen Task.
Das Argument "s:Startup-Sequence" sorgt dafür, daß die gewünschte Batchdatei automatisch angezeigt wird. Nachdem Sie fehlerfrei abgetippt haben, speichern Sie mit "Save As" die Datei als "s:Startup-Sequence" ab und starten den Computer erneut (Versuchen Sie mal, mit einer Hand einen Reset zu machen!).

```
ECHO "Das ist meine neue
                 StartupSequence"
LOADWB
SYS:System/Setmap d
RUN SYS:Utilities/Notepad
s:Startup-Sequence
```

Schon wieder im Notepad? Wenn ja, haben Sie gesehen, was in der Startup-Sequence so alles gemacht wurde?

Nun gehen wir etwas weiter. Wir wollen in der Startup-Sequence auch noch die Shell starten. Diese Shell ist allerdings erst unter der Workbench 1.3 verfügbar. Hierzu geben Sie Startup-Sequence 2 ein. Die Installation, die diese Startup-Sequence durchführt, setzen wir für den Rest des Artikels als gegeben voraus. Auf die Bedeutung und Auswirkung dieser Installation für Batchdateien gehen wir später nochmal ein. Im folgenden gehen wir auch nicht mehr auf die Benutzung des Notepads oder des Editors "ED" ein. Sie sollten diese Schritte selbst beherrschen, und wenn nicht, Probieren geht am AMIGA wirklich über Studieren.

sollte man sinnvollerweise wie folgt vorgehen im RAM: ein Directory "c" erzeugen und in dieses Directory die gewünschten CLI-Befehle kopieren. Anschließend noch den Pfad auf dieses Directory legen, damit der Rechner die CLI-Befehle im RAM: findet.

Die in Batch_1 gezeigte Version zur Erstellung einer RAM-Disk ist sehr einfach, aber auch unelegant und langsam. Dies hat u.a. den Grund, daß der COPY-Befehl für jeden Kopiervorgang neu von Diskette geladen werden muß. Um das zu verhindern, sollte man den COPY-Befehl als ersten ins RAM: kopieren und von da aus aufrufen. Eine Batchdatei, die nach diesem Schema vorgeht, sieht man in dem File Batch_2.

Aber auch das läßt sich noch verbessern, denn der COPY-Befehl kennt die Möglichkeit des ODER-Kopierens. Das

```
ECHO "Das ist meine neue Startup-Sequence"
PATH ram: c: :system:utilities add        ; Pfad für CLI festlegen
LOADWB                                    ; Workbench starten
Setmap d                                  ; deutsche Tastaturbelegung
RESIDENT CLI L:Shell-Seg SYSTEM PURE ADD  ; Shell aktivieren
MOUNT newcon:                             ; Starten des nuen
Consoledevices
NEWSHELL                                  ; Shell starten
ENDCLI >NIL:                              ; Start-CLI wird geschloßen
```

*Startup-Sequence 1: Systemstart*

```
MAKEDIR        ram:c          ; Directory c im RAM: erzeugen
COPY c:dir     ram:c          ; Dir ins RAM:c kopieren
COPY c:cd      ram:c          ; Cd ins RAM:c kopieren
COPY c:list    ram:c          ; List ins RAM:c kopieren
COPY c:copy    ram:c          ; Copy ins RAM:c kopieren
COPY c:delete  ram:c          ; Delete ins
RAM:c kopieren
COPY c:makedir ram:c          ; Makedir ins RAM:c kopieren
PATH ram:c add                ; Zusätzlich Pfad auf RAM:c legen
```

*Batch_1: Einfacher aber unelegant und langsamer Batch*

```
MAKEDIR                ram:c   ; Directory c im RAM: erzeugen
COPY c:copy            ram:c   ; Copy ins RAM:c kopieren
RAM:c/copy c:dir       ram:c   ; Dir ins RAM:c kopieren
RAM:c/copy c:cd        ram:c   ; Cd ins RAM:c kopieren
RAM:c/copy c:list      ram:c   ; List ins RAM:c kopieren
RAM:c/copy c:delete    ram:c   ; Delete ins RAM:c kopieren
RAM:c/copy c:makedir   ram:c   ; Makedir ins RAM:c kopieren
PATH ram:c add                 ; zusätzlich Pfad auf RAM:c legen
```

*Batch_2: verbesserte. aber immer noch nicht die wahre Version*

```
copy c:dir|cd|copy|list|delete|makedir ram:c
path ram:c add
```

*Batch_3: Die schnellste Methode des Kopiervorhabens*

## Batchdatei erstellt RAM-Disk

Batchdateien sollen dem Anwender Arbeit abnehmen. Eine typische Arbeit ist das Einrichten einer RAM-Disk. Um mit einer RAM-Disk arbeiten zu können, muß sich im Directory "L" der Workbench das File RAM-Handler befinden. Beim Erstellen einer RAM-Disk

bedeutet, daß man mehrere Files angeben kann, und der der Copy-Befehl probiert, alle zu kopieren. Dies kann man ausnutzen, um gezielt verschiedene Files zu kopieren. Wie das funktioniert, sieht man in Batch_3. Dortwird automatisch das Directory "c" in RAM: erzeugt. Der COPY-Befehl erzeugt ein nicht vorhandenes Ziel-Directory immer dann, wenn man mehrere Files (mittels der ODER-Funktion oder eines Musters) angibt. Falls man nur "COPY c:dir ram:c" schreibt, erzeugt der COPY-Befehl im RAM: ein File (kein Directory!) mit dem Namen "c". Falls sich im RAM schon ein Directory "c" befindet, wird der DIR-Befehl in dieses Directory kopiert. Batch_3 hat den unbestreitbaren Vorteil, daß der COPY-Befehl nur einmal geladen wird. Man spart sich weiterhin auch die Ladezeit für den MAKEDIR-Befehl. (Verdammt kurz geworden, oder?)

Was wir bis jetzt erfahren haben, ist, wie man Batchdateien eingibt und wie man sie optimieren kann, aber wir wissen noch nicht, wie man eine Batchdatei startet, wenn man von der Startup-Sequence absieht.

Zum Starten von Batches gibt es zwei Methoden. Fangen wir mit der einfachen an, dem EXECUTE-Befehl. Dies ist die (bisher) übliche Methode, um eine Batchdatei ausführen zu lassen. Der Aufruf für den EXECUTE-Befehl sieht wie folgt aus:

```
EXECUTE Name [Parameter]
```

z.B. `EXECUTE BATCH_1`

Jetzt sollte man mal an die Arbeit gehen und die Batchdateien Batch_1 bis Batch_3 eingeben und ausprobieren. Kommen wir nun zu der zweiten Art, eine Stapeldatei zu starten. Diese Methode hängt eng mit der unter der WB 1.3 verfügbaren Shell zusammen. Wir haben ja oben erwähnt, daß wir noch einmal auf die von uns besprochene Startup-Sequence 2 zurückkommen. Nun ist es soweit.

## Automatisches Starten von Batchdateien

In der Startup-Sequence wird das Programm Shell-Seg resident gemacht. Die Kommandozeile dafür sieht wie folgt aus:

```
RESIDENT CLI L:Shell-Seg SYSTEM
                           PURE ADD
```

Das Shell-Seg-Programm, das sich im Directory "L" der Workbench befindet, wird unter dem Namen CLI resident gemacht. Die Option SYSTEM bedeutet, daß das "CLI" nur durch das Betriebssystem angesprochen werden kann. Das Flag PURE (siehe File-Flags und ihre Bedeutungen) veranlaßt den RESIDENT-Befehl den Status des Pure-Flags (pure = rein, gemeint ist reiner, sauberer Code) zu ignorieren. An sich soll man nur Programme resident machen, deren Pure-Flag gesetzt ist. Falls man ein Programm resident macht, welches nicht entsprechend programmiert ist, kann man davon ausgehen, daß die Angelegenheit in die Hose geht, im schlimsten Fall grüßt der GURU. Die Option "ADD" fügt Shell-Seg zu der bestehenden Residentliste hinzu, dieses Flag ist nicht unbedingt nötig.

Nachdem man das "Shell-Seg" resident gemacht hat muß noch eine neue Shell gestartet werden. Erst in dieser Shell findet die Auswertung des Shell oder Script-Flags (siehe File-Flags und ihre Bedeutungen) statt. Falls dieses Flag bei einer Datei gesetzt ist, wird sie automatisch ausgeführt, sobald wir ihren Namen eingeben, so als wenn wir sie mit dem EXECUTE-Befehl gestartet hätten.

Die Anweisung "Mount Newcon:" hat mit diesem Vorgang nichts zu tun, "NewCon" ist nur für die Verwaltung der Tastatur zuständig und damit u.a. auch für die unter 1.3 mögliche Benutzung der Cursor-Tasten. In der Praxis sieht die Anwendung des Shell-Flags wie folgt aus: Zum Setzen oder Löschen des Flags benutzt man den PROTECT-Befehl.

```
PROTECT Name {Flag} (Add|Sub)
```

Beispiel: `PROTECT Batch_1 S ADD`

In diesem Beispiel wird das Shell-Flag des Files "Batch_1" gesetzt. Wenn man nun Batch_1 eingibt, wird die Datei genauso gestartet, als ob man "EXECUTE Batch_1" eingegeben hätte. Es sei hier noch mal festgehalten, daß sich bei gesetztem S-Flag der EXECUTE-Befehl erübrigt. Die Datei läßt sich starten, ohne daß man sie mit "EXECUTE Name" aufruft. Bei jedem Kopiervorgang werden die File-Flags mitkopiert. Nur falls man die Datei mit einen Texteditor neu schreibt, wird das Flag gelöscht.

Beispiel:

```
Batch_1 ; S-Flag der Datei sollte
          gesetzt sein
```

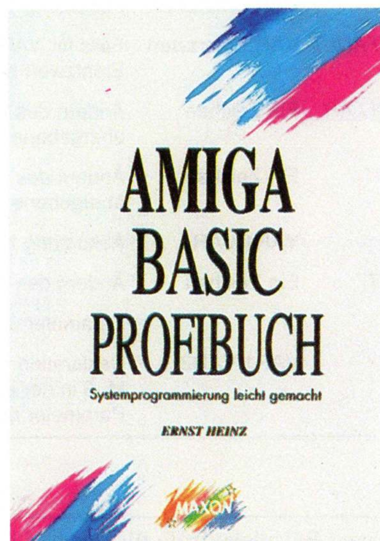
## Parameter und die Batchdatei

Die bisherigen Stapeldateien konnte man nur starten. Eine Beeinflussung des Ablaufes war nicht möglich. Wir kommen jetzt zu Batchdateien, die eine Beeinflussung in Form von Parametern kennen. Parameterübergabe an Programme kennt sicher jeder, denn wie sonst könnte man einem COPY-Befehl

### File-Flags und ihre Bedeutungen

| Flag | Flag-Name | Bedeutung |
|---|---|---|
| S | Shell-Flag | automatisches Starten als Batchdatei |
| P | Pure-Flag | File läßt sich resident machen |
| A | Archive-Flag | Von dem File existiert ein Backup. |
| R | Read-Flag | Datei kann gelesen werden |
| W | Write-Flag | Datei kann geschrieben werden |
| E | Execute-Flag | File kann gestartet werden<br>In der Literatur findet man auch die Übesetzung Edit-Flag, was allerdings Unsinn ist.<br>Ein normales CLI unterstützt dieses Flag allerdings nicht. |
| D | Delete-Flag | File kann gelöscht werden |

# AMIGA BASIC Profibuch




NEU!

## Für Theoretiker

Dieses Buch gibt Ihnen einen hervorragenden Überblick über die wichtigsten Möglichkeiten und Methoden der Systemprogrammierung unter AMIGA-BASIC. Neben einer detaillierten Beschreibung und Analyse der wichtigsten Funktionen finden Sie in diesem Buch eine ausführliche Erläuterung der verschiedensten System-Unterroutinen sowie den Aufbau des Gesamtsystems.

## Für Praktiker

Im Hinblick auf die praktische Anwendung in eigenen Programmen dürfte die Sammlung von über 60(!) neuen AMIGA-BASIC-Unterroutinen zur Systemprogrammierung ein absoluter Lekkerbissen für jeden BASIC-Programmierer sein. Mit diesen neuen Unterroutinen werden die fantastischen Möglichkeiten des AMIGA nun endlich auch all denjenigen in einfacher Weise zugänglich gemacht, die selbst nicht tiefer ins Sytem eindringen möchten.

## Für jeden

Blitzschnelle und variationsreiche Textausgabe, die Benutzung beliebiger Textfonts, die Erzeugung von Pull-Down-Menüs, die Gestaltung eigener Requester sowie die komplette Fenstersteuerung werden ab sofort für jeden BASIC-Programmierer leicht realisierbar.

Aber noch viel mehr Geheimnisse erwarten Sie während der Lektüre der einzelnen Kapitel.




Bestellcoupon MAXON Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Name: ______________________

Vorname: ______________________

Straße: ______________________

Ort: ______________________

Unterschrift: ______________________

Hiermit bestelle ich:

☐ Exemplar(e) von "AMIGA BASIC Profibuch". Mit Diskette für DM 59,00

☐ Vorauskasse

☐ Nachnahme

Versandkosten: DM 7,50
Nachnahme zuzgl. DM 3.80 Nachnahmegebühr.

Vertrieb in der Schweiz: DTZ DataTrade AG Langstrasse 94 Postfach 413 CH-8021 Zürich Tel.: 01/242 80 88 Fax.: 01/291 05 07
Vertrieb in Österreich: Dipl.-Ing. Reinhart Temmel Ges.m.b.H. & Co.KG Markt 109 A-5440 Golling Tel.: 06244/7081-17 Fax.: 06244/7188-3
Vertrtieb in Holland: JOTKA COMPUTING Postbus 8183 NL-6710 AD Ede Tel.: 08380/38731 Fax.: 08380/21675

mitteilen, was er zu kopieren hat? Die Parameter bei Batchdateien funktionieren genau so. Ein typischer Aufruf für eine Batchdatei mit Parametern könnte wie folgt aussehen:

```
EXECUTE Batch_4 Carsten Heike
```

Falls das Shell-Flag gesetzt ist, ist der EXECUTE-Befehl unnötig. Um eine Batchdatei dazu zu bringen, Parameter zu akzeptieren, muß die erste Zeile der Batchdatei wie folgt aussehen:

```
.key VAR1,VAR2,...
```

Eine typische Anwendung des .KEY-Befehls wird in Batch_4 an einem Beispiel gezeigt. Die Aufgabe der Batch_4-Datei besteht darin, die Namen zweier Dateien zu tauschen. Die Parameterübergabe findet in der ersten Zeile statt. Die Variablen nennen sich Name1 und Name2. Zur Kontrolle kann man Textausgaben, die auch die Parameter ausgeben, einbinden. In Batch_4 wird das auch demonstriert. Um eine Variable in einer Batchdatei ansprechen zu können, muß man den Namen der Variablen in die Zeichen “<.....>” einschließen. Die Zeichen “<“ und “>” kann man mit dem Befehl “.BRA” bzw. “.KET” ändern. Falls man bei der Variablendeklaration an den Namen der Variablen ein “/a” anhängt, bedeutet dies, daß man diesen Parameter in jedem Fall übergeben muß, weil sonst die Stapel-Datei nicht gestartet wird.

Falls man an einen Variablennamen ein “/k” anhängt, muß beim Aufruf der Batchdatei der Parameter immer von seinem Namen und einem / eingeleitet werden. Was das bedeutet, kann man in Batch_5 sehen. Als letztes Anhängsel gibt es noch “/s”. Die Möglichkeit, die dieser Parameter bietet, ist die folgende: Man kann für diesen Parameter nur ein festgelegtes Wort übergeben - den Namen, den man als Variable angegeben hat.

Falls man für einen Parameter nichts übergibt, kann man eine Batchdatei dazu veranlassen, einen sogenannten Default-Wert zu benutzen. Dazu gibt es zwei Möglichkeiten: zum einem mit dem .DEFAULT-Befehl und zum anderen mit dem Dollarzeichen “$” (kann mit “.DOLLAR” geändert werden). Wie das Dollarzeichen angewandt wird, sehen wir in Batch-5. Die Anweisung “<Wert$30>” veranlaßt den EXECUTE-Befehl, für die Variable “Wert” “30” zu benutzen, falls man der Variablen “Wert” keinen Parameter übergeben hat. Die andere Möglichkeit mit dem .DEFAULT-Befehl wird auch in Batch_5 gezeigt. In der zweiten Zeile wird der Variablen “Normal” der Text “ram:” zugeordnet. Falls “Normal” beim Aufruf der Batchdatei kein Parameter übergeben wurde, erhält die Variable “Normal” den Wert “ram:”.

## Punkt-Befehle und ihre Bedeutungen

| *Befehl* | *Parameter* | *Bedeutung* |
|---|---|---|
| *.BRA* | Ein Zeichen | Ändern des Kleinerzeichens in das als Parameter übergebene Zeichen. “.BRA {“ ändert “<“ in “{“. |
| *.DEF* | VAR,Ersatzwert | Abkürzung für .DEFAULT-Befehl |
| *.DEFAULT* | VAR,Ersatzwert | Falls für VAR kein Wert übergeben wurde, wird der Ersatzwert benutzt. |
| *.DOLLAR* | Ein Zeichen | Ändern des Dollarzeichens in das als Parameter übergebene Zeichen. “DOLLAR &” ändert “$” in “&”. |
| *.DOT* | Ein Zeichen | Ändern des Punktzeichens in das als Parameter übergebene Zeichen. “.DOT %” ändert “.” in “%”. |
| *.K* | VAR1,VAR2,... | Abkürzung für .KEY-Befehl |
| *.KET* | Ein Zeichen | Ändern des Größerzeichens in das als Parameter übergebene Zeichen. “.KET }” ändert “>” in “}”. |
| *.KEY* | VAR1,VAR2,... | Deklaration der Übergabeparameter an eine Batchdatei. Muß in der ersten Zeile der Datei stehen. Falls ein Parameter übergeben werden muß, hängt man an die |

Ein paar Beispielaufrufe für Batch_5 könnten wie folgt aussehen:

```
1. EXECUTE Batch_5 Programm Option1 Wert=500
2. EXECUTE Batch_5 Programm df0:    Wert=500
3. EXECUTE Batch_5 Programm df1:    Wert 500 Option1
4. EXECUTE Batch_5 Programm
```

Die Ausgabe der Batchdatei sieht dann folgendermaßen aus:

```
1. compile Programm.pas to ram:Programm.obj Option1 Größe 500
2. compile Programm.pas to df0:Programm.obj  Größe 500
3. compile Programm.pas to df1:Programm.obj Option1 Größe 500
4. compile Programm.pas to ram:Programm.obj  Größe  30
```

```
.KEY Name1/a,Name2/a          ; Parameter einlesen
ECHO "Aus dem File <Name1> wird <Name2>."
ECHO "Aus dem File <Name2> wird <Name1>."
RENAME <Name1> <Name1>.merk   ; Name1 wird in Name1.merk umbenannt
RENAME <Name2> <Name1>        ; Name2 wird in Name1 umbenannt
RENAME <Name1>.merk <Name2>   ; Name1.merk wird in Name2 umbenannt
```

*Batch_4: Umbenennung zweier Dateien*

```
.KEY file/a,Option1/s,Normal,Wert/k
.DEFAULT Normal ram:
ECHO "compile <file>.pas to <Normal><file>.obj <Option1> Größe <Wert$30>"
```

*Batch_5: Auch beim Batchen spielt der Dollar eine Rolle*

## Batchdateien in der Praxis

Nachdem wir Ihnen die Grundlagen von Batchdateien nahegebracht haben, wollen wir uns noch mit einigen Anwendungen von Batchdateien beschäftigen.

Zuerst wollen wir ein Batch-File programmieren, das es uns ermöglicht, festzuhalten, wann wir den Rechner eingeschaltet und was wir getan haben. Dieses Batch-File soll von anderen Batch-Files oder vom CLI aus aufgerufen werden können. Als Parameter soll die Tätigkeit übergeben werden, die wir ausführen möchten. Falls wir keinen Parameter übergeben, soll "Workbench&CLI" in das Logfile eingetragen werden. Damit das Batch-File ordnungsgemäß abläuft, ist das Vorhandensein einer Datei mit dem Namen Logfile im Device/Directory s: notwendig. In diese Datei (Logfile) kann man einfach "Mein Logfile" eintragen.

In der Stapeldatei wird der ECHO-Befehl auf eine andere als bisher übliche Art verwendet. Denn bis jetzt schrieb der ECHO-Befehl alle Ausgaben direkt in das CLI-Window. Mit dem Umleitungszeichen ">" kann man die Ausgaben eines beliebigen Befehles in eine Datei umlenken. Bei dem im Make_LogFile verwendetem Aufruf des ECHO-Befehls wird also im RAM: eine Datei mit dem Namen "Log1" erzeugt. In diese Datei wird als erstes der Text "Am " geschrieben. In der nächsten Zeile passiert wieder etwas Neues. Das Umleitungszeichen ">" wurde schon erklärt. Falls man zwei Umleitungszeichen, also ">>" verwendet, bedeutet dies, daß man die Ausgabe an ein bereits bestehendes File anhängen möchte. Man beachte, daß diese Möglichkeit nur in der WShell funktioniert, für die Leute, die diese Shell nicht besitzen, haben wird uns auch etwas ausgedacht. Make_Logfile2 macht das gleiche wie Make_Logfile, nur benötigt es nicht die WShell zum Laufen.
Folglich wird die Ausgabe des DATE-Befehls an "Log1" angehängt. Das gleiche passiert mit dem nächsten ECHO-Befehl. Der JOIN-Befehl fügt nun die Datei "Log1" mit der Datei "LogFile" zu der Datei RAM:LogFile zusammen. Diese Datei wird wieder nach s:LogFile kopiert, und damit wird das alte LogFile überschrieben. Mit DELETE werden aus dem RAM: die zwei Files (Log1 und LogFile) gelöscht. Die Ausgabeumleitung zu dem Null-Device NIL: bedeutet, daß die Ausgaben des DELETE-Befehls nicht sichtbar werden.

```
.KEY Was
.DEFAULT Was Workbench&CLI
ECHO  >ram:Log1 "Am "
DATE  >>ram:Log1
ECHO  >>ram:Log1 " wurde am Rechner mit <Was> gearbeiten"
JOIN  ram:Log1 s:LogFile as ram:LogFile
COPY  ram:LogFile s:
DELETE >NIL: ram:Log1|LogFile
```

*Make_LogFile*

```
.KEY Was
.DEFAULT Was Workbench&CLI
ECHO  >ram:Log1 "Am "
DATE  >ram:Log2
ECHO  >ram:Log3 " wurde der Rechner gestartet, um mit <Was> zu arbeiten"
JOIN  ram:Log1 ram:Log2 ram:Log3 s:LogFile as ram:LogFile
COPY  ram:LogFile s:
DELETE >NIL: ram:Log1|Log2|Log3|LogFile
```

*MakeLogFile2:*

## Schnelle Ausgabe im CLI

Ein häufig auftretendes Problem ist die Ausgabe von Informationen oder Hilfestellungen. Üblicherweise geschieht das mit Hilfe des ECHO-Befehls.
Das ist allerdings eine recht langsame und umständliche Methode. Wenn man einmal vor diese Aufgabe gestellt ist, sollte man sie mit dem TYPE-Befehl lösen. Den gewünschten Text in eine kleine Datei geschrieben, und in der Batchdatei schreibt man dann nur noch "TYPE DateiName", und schon ist der Text sichtbar.

## Schnelle Ausgabe am Drucker

Um eine Überschift über ein Listing zu bekommen, oder um mal schnell ein paar Zeilen auszudrucken, ist es doch recht aufwendig, erst mal WordPerfect zu laden. Das kann man auch einfacher haben. Mit der Zeile "COPY * PRT:" kann man mal eben die Tastaturausgabe zum Drucker schicken. Wenn man ewCon: oder eine andere Shell laufen hat, kann man in der eingegebenen Zeile sogar noch komfortabel edieren, bevor man sie mit RETURN ausdruckt. Der Vorgang wird mit CTRL-C abgebrochen. Im Batchfile COPY-PRT sieht man noch einmal eine komplette Lösung.

## Schneller Abbruch

Eine gestartete Batchdatei kann man auch abbrechen. Das geschieht mit CTRL-D. Vielfach liest man, daß dafür die Tastenkombination CTRL-C gedacht sei, was aber falsch ist. Mit CTRL-C kann man, soweit möglich, einen Befehl abbrechen. Es ist damit aber prinzipiell nicht machbar, eine Batchdatei abzubrechen. Wenn dies doch geschieht, hängt das mit dem sogenannten RETURN-Code des abgebrochenen Befehles zusammen. Doch das erfahren Sie erst im nächsten Teil. ■

```
ECHO " Eine einfache Schreibmaschine ... "
ECHO " Bitte Text eingeben, mit CTRL-C kann man abbrechen"
copy * prt:
COPY-PRT: Simultion einer Schreibmaschine
```

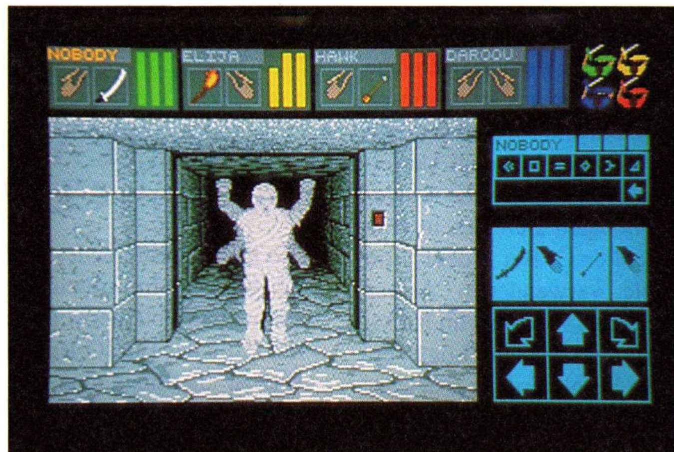
*Eine einfache Schreibmaschine*

VON NICOLAS HEIDE

# Tips zu
# Dungeon Master

*Ziel des Spiels ist es, den Grey Lord bei seinem Streben nach Frieden und Freiheit im fernen Lande Viborg zu unterstützen. Dazu muß man einen magischen Edelstein, den Power Gem, aus dem Dungeon, einem gewaltigen Höhlensystem, entwenden. Doch dieser Stein barg herbe Überraschungen und richtete sich gegen seinen Herrn, worauf dieser ihn verlor.*

Mutige Abenteurer versuchten sich schon daran, sind jedoch gescheitert. Nun ist es Ihre Aufgabe, den Power Gen zu finden. Dazu haben wir, um Ihnen die Arbeit zu erleichtern, einige wichtige Tips zusammengetragen.




### Die netten lila Würmer:

Nuja. Diese fiesen Viecher sollte man auf keinen Fall aus der Nähe attackieren, da sie verflixt hart und vergiftend zurückschlagen.
Zudem: Wozu hat man denn seine Magier? Mit einigen FUL IR (ca. vier mittlerer Stärke) steigt nicht nur deren Magic-Skill, sondern auch die Furcht der Würmer.

### Wandelnde Schutthaufen:

Um diese wirkungsvoll zu bekämpfen hat man zwei Möglichkeiten:
1. viele schwache FUL IR-Spells.
2. die Magier(!) in die erste Reihe stellen. Nun schlagen (nicht treten!), was die Fäuste hergeben.
Methode zwei bewirkt normalerweise mindestens einmal die Erhöhung des Ninja-Skills.

### Die übergroßen Fliegen:

Falls keine Fliegenklatschen zur Hand: Zwei FUL IRs sind oft ausreichend. Ein Mann-gegen-Fliege-Schwertkampf dauert wesentlich länger.

### Die Drachen:

Allem Getöse zum Trotz sollte man im Kampf ruhig und konzentriert bleiben, da der Drache oft plötzlich und schnell die Position wechselt. Feuerbälle, Shurikens, Messer, Hiebe und Schläge; alles ist erlaubt. Gute Schwertkämpfer sind Gold wert!!

### Die grünen Geister:

Ein DES EW Spell reicht meist aus. Man findet Staffs, mit denen man den "Dispell" casten kann; spätestens dann stellen sie kein Problem mehr dar. Mit Schwertern, Äxten o.ä. bringt man sie höchstens zum Lachen.

### Der stehlende Zwerg:

Er stellt keine Gefahr dar, klaut aber Items, die man in der linken Hand hält. Spells wirken nur, wenn er sich der Party nähert, beim Weglaufen weicht er Feuerbällen meist aus. Gestohlene Gegenstände verliert er wieder, wenn man ihn erwischt hat.

### Die quabbligen Wizards:

Sie sind äußerst gefährlich. Verschanzt man sich hinter einer Tür, casten sie oft ZO-Spells (die Tür öffnet sich!). Also Vorsicht, und immer zwei fröhliche FUL IR auf den Lippen.

### Die zähen Ratten:

Um diese zu besiegen, braucht man schon einige Zeit, da sie anscheinend ein sehr dickes Fell besitzen. Hat man noch genug Mana für einige Spells, sollte man es verwenden.

### Die Riesenskorpione:

Oha. Rennen oder Casten sind die einzigen Möglichkeiten die man hat. Doch Vorsicht! Um sie zu eliminieren, braucht es schon fünf starke FUL IR. Da die Skorpione hemmungslos kräftig zuschlagen, überlebt man den Nahkampf nur sehr selten.

### Die Schlangen:

Zum Flüchten ist es meistens schon zu spät, wenn man auf Tuchfühlung ist. Wenn nicht zwei starke DES EW parat sind, braucht man kaum gegen sie anzutreten.

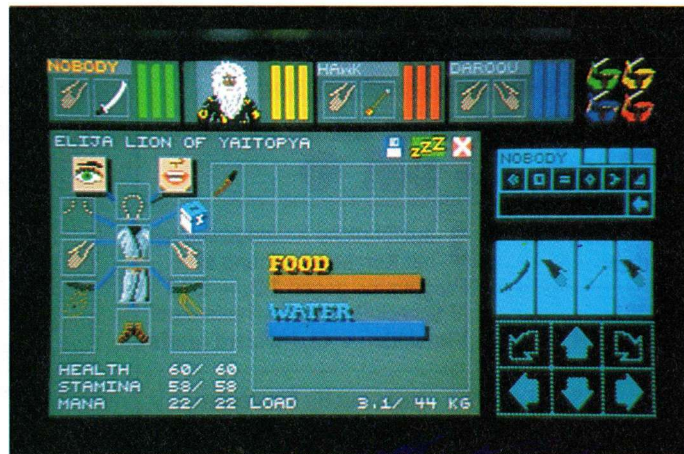
### Die zweischwertigen Ritter:

Sie sind zwar nicht so gefährlich wie die Skorpione, schlagen aber recht satt zu. Da der Ritter sehr träge ist, bietet sich die "hit and run"-Kampftaktik an; es dauert aber geraume Zeit, bis man einen geplättet hat.

### Skelette, Mumien, Slimes und blaue Trolls:

Bei diesen Kreaturen reicht oft eine starke Tür aus, die man versucht, auf ihrem Kopf zu zerbrechen. Findet man beim Kampf Dinge, die sich als Wurfgeschosse eignen, sollte man diese den Viechern entgegenschleudern; das wirkt oft mehr als viele Schwertschläge.

Viel Erfolg




## Einige Tips für die eigenen Kämpfer:

1.) Alle Abenteurer sollten fähig sein, Zaubersprüche anzuwenden.

2.) Schon vom ersten Level ab: Zaubern und brauen, was das Zeug hält. Besonders den Spell FUL IR möglichst schnell lernen.

3.) Priester können mit genug Erfahrung zu Magiern werden.

4.) Damit die zweite Reihe beim Kampf nicht nutzlos herumsteht, empfiehlt es sich, diese mit Shurikens und Messern auszustatten, welche dann geworfen werden können.

5.) Flaschen, in denen sich Potions befanden, nicht wegwerfen; auch wenn Euer Priester später ca. 12 Stück mit sich rumträgt.

6.) Ab dem vierten Level sollte man immer zwei VI BRO-Potions bei sich haben, da die Monster nun recht giftig werden.

7.) Viele Monster ergreifen kurz vor ihrem Exitus die Flucht. Folgt man ihnen, verläuft man sich oft in den Tiefen der Dungeons.

8.) Eine große Rolle spielt das Gesamtgewicht der Kämpfer. Wormrounds sollte man möglichst schnell essen, da sie kaum nahrhaft, dafür aber extrem schwer sind.

9.) Nach dem 2. Level findet man lange kein Wasser mehr. Hat man das 3. Level durchgespielt, sollte man sich nochmal im 2. Level mit Wasser versorgen.

10.) Im 4. Level ist ein Raum voller fieser Slimes. Hier kann man nicht nur seine Skills verbessern; jeder Slime hinterläßt als Erbe mehr oder weniger schmackhaftes Fleisch. Entfernt man alle Slime-Überreste aus dem Raum, trifft man nach ein paar Minuten schon wieder auf die nächste Generation.

11.) Oft stößt man auf Altäre, die mit dem Slogan "New Live For Old Bones" für sich werben. Hat ein Monster einem unserer Helden das Lebenslicht ausgeblasen, kann man den Charakter hier verlustlos wiederbeleben lassen.

12.) Zaubert man Spells gegen eine Wand, um seine Skill-Werte zu erhöhen, wird man kaum Erfolg haben. Läßt man aber den Party-Leader einige Ausrüstungsgegenstände durch die Dungeons werfen, so erhöht sich schon bald sein Ninja-Level.

VON FRANK SCHÄFER

# *MUSCHEL MIT PERLE?*

## *Das Programmiertoll WShell*

*William S. Hawes ist schon durch das beliebte Programm "Conman" bekannt geworden. Letztes Jahr brachte er eine Shell mit dem Namen "WShell" auf den Markt, von der jetzt als Update die Version 1.05 erschienen ist. Grund genug für uns, dies in Programiererkreisen sehr beliebte Tool anzuschauen.*

```
1627128 Work:Assembler/Programme/Programme-Arbeit/Laufschrift
R(0);6:37PM> dir env:
  echo                                   echo.info
  path                                   path.info
  qw2                                    shellwindow
  shellwindow.info                       titlebar
  titlebar.bak                           titlebar.bak.info
  titlebar.info
R(0);6:38PM> alias
CLone    = copy [] clone
cls      = echo "*e[0;0H*e[J"
DELete   = delete
FLush    = stack >nil: 12000000
HIStory  = history
LIst     = list
MAkeDir  = makedir
MKdir    = makedir
PRotect  = protect
Quick    = Format drive [] Name Leer Quick
REname   = rename []
Run      = runwsh
tb       = runwsh turbobackup []
troff    = echo >env:echo OFF
tron     = echo >env:echo ON
R(0);6:38PM>
R(0);6:38PM>
R(0);6:38PM>
```

Das Environment-Device der WShell

Die Allias-Befehle der WShell

Der Prompt der WShell mit Returncode und Uhrzeit

In der Titelzeile wird Speicher und Pfad angezeigt

Zunächst aber erst einmal ein paar grundlegende Informationen über Shells. Unter einer Shell versteht man ein Programm, das die Schnittstelle zwischen dem Betriebssystem und dem Benutzer bildet. Dieses Programm soll es dem Benutzer möglichst leicht machen, mit den Betriebskommandos umzugehen. Der Alias-Befehl, die Funktionstastenbelegung und die Benutzung der Cursortasten sind übliche Möglichkeiten einer Shell.

Beginnen wir bei der Beschreibung der WShell mit dem mitgelieferten Handbuch. Das Handbuch umfaßt 61 kleingedruckte DIN A5-Seiten und besitzt damit für eine Shell einen wirklich ausreichenden Umfang. Wir hatten von dem Handbuch bzw. der WShell zwei Versionen im Test; zum einen die amerikanische Version und zum anderen eine gut gelungene Übersetzung der Firma Compustore in Frankfurt. Das amerikanische Handbuch ist für jemanden mit normalen Schulenglischkenntnissen gut zu verstehen und bietet alle notwendigen Informationen, die zum Umgang mit der WShell notwendig sind.
Seltsamerweise ist die Schnittstelle zu ARexx, welches ja auch ein Produkt von William S. Hawes ist, für einen Anfänger nicht genügend umfangreich dokumentiert. Von der deutschen Version des Handbuches lag uns leider nur eine Vorabversion vor, so daß wir über die endgültige Qualität des Produktes keine Aussage machen können.

## Muschel mit Variablen

Wenn Sie das Fotos der WShell betrachten, werden Sie einen wesentlichen Unterschied zum CLI feststellen. In der

Titelzeile des WShell-Windows sieht man, wieviel Speicherplatz zur Zeit frei ist und in welchem aktuellen Directory man sich befindet. Weiterhin sind im Prompt der WShell der Fehlercode (Returncode) des letzten Kommandos und die aktuelle Uhrzeit sichtbar. Diese Dinge kann man mit Hilfe der Statusvariablen und des ENV:-Devices (Environment, zu deutsch so viel wie Umgebung) einstellen. In diesem Device gibt es einige ASCII-Dateien, die wie Variablen verwendet werden. Mit der Echo-Variablen/-Datei kann man zum Beispiel angeben, ob alle Befehle ausführlich mitprotokolliert werden sollen. Wenn man das Wort "ON" in diese Datei schreibt, wird zum Beispiel jeder Alias-Befehl ausgeschrieben oder der Ablauf einer Batchdatei zeilenweise angezeigt. Falls man "OFF" in diese Datei einträgt, passiert nichts von dem. Mit der Datei "Titlebar" hat es eine ähnliche Bewandtnis.
In ihr kann man mit Hilfe einiger Statusvariablen (siehe Tabelle) angeben, welcher Text in der Titelzeile stehen soll. Schreibt man in die Datei Titelzeile den Text "*%L%T%G Pfad %C*", sieht die Titelzeile wie folgt aus:

"<19:23:43> Pfad df0:utilities".

## Muschel mit anderen Fenstern

Nicht nur in Beziehung auf die Statusvariablen hat die WShell dem Benutzer einige Neuigkeiten zu bieten. Auch die Windows der WShell sind wesentlich flexibler als die des Standard-CLIs. Falls man eine WShell auf folgende Art startet: "*NewWSh con:0/0/200/100/Name/acn*", erhält man ein Window, das beim Start aktiviert ist, ein CloseGadget besitzt und keinen Windowrahmen (Border) hat. Außer den drei schon beschriebenen Optionen gibt es noch die Parameter *b,d,m,r* und *s*. Viel Spaß beim Ausprobieren.

## Muschel mit Funktionstasten

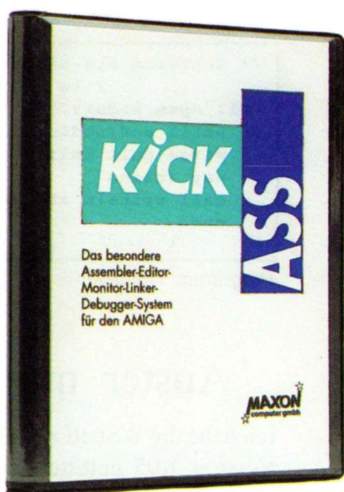
Die Tastaturbelegung unterscheidet sich erheblich von der normalen des CLI. In nebenstehender Tabelle sieht man, welche Möglichkeiten die WShell bietet.

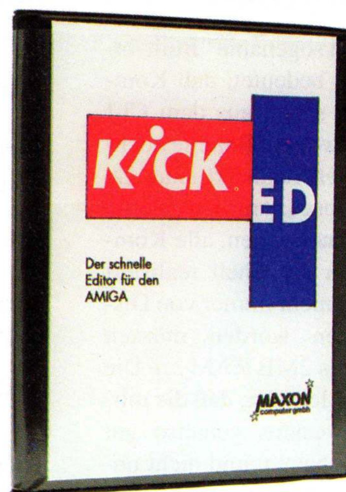
| *Variable* | *Titelzeile/Prompt* | *Erläuterung* |
|---|---|---|
| %B | T P | senkrechter Strich "\|" |
| %C,%c | T P | aktuelles Directory |
| %E | T P | verstrichene Zeit |
| %F | T P | Fehlerabbruchlimit |
| %G | T P | Größerzeichen ">" |
| %I | T P | Adresse des aktuellen Tasks |
| %L | T P | Kleinerzeichen "<" |
| %mc,%MC | T P | Chip Memory (Bytes oder K Bytes) |
| %mf,%MF | T P | Fast Memory (Bytes oder K Bytes) |
| %mp,%MP | T P | Public Memory (Bytes oder K Bytes) |
| %N | T P | Nummer des aktuellen Tasks |
| %P | T P | Task-Priorität |
| %R | T P | Rückgabewert des letzten Befehles |
| %S | T P | Größe des Stacks |
| %t,%T | T P | Uhrzeit 24/12-Format |
| %W | T | Original Windowtitel |
| %Y | T P | Fehlercode |
| %1-%7 | P | Wählen einer bestimmten Farbe |
| %[Programm] | P | Programm wird bei jeden Erscheinen des Promptes gestartet |
| *N | P | neue Zeile |

| *Taste* | *Funktion* |
|---|---|
| CTRL-A | Wechseln zwischen Überschreib- und Einfügemodus |
| CTRL-B | Löschen des History-Buffers |
| CTRL-Q | gestoppte Ausgabe fortsetzen |
| CTRL-R | Wiederholung der obenstehenden Zeile |
| CTRL-S | Ausgabe stoppen |
| CTRL-U | Zeile links des Cursors löschen |
| CTRL-W | Window neu aufbauen |
| CTRL-X | gesamte Eingabezeile löschen |
| CTRL-Y | Zeile rechts des Cursors löschen |
| CTRL-Z | lösche die oben geschriebene Zeile. |
| CTRL-] | springe zum Anfang/Ende der Zeile. |
| CTRL-^ | Wähle den Einfügemodus an. |
| CTRL— | Lade von Undo-Buffer. |
| CTRL-\ | Sende END OF FILE ans CLI. |
| Cursor nach oben | Historyliste eins nach oben |
| Cursor nach unten | Historyliste eins nach unten |
| Cursor nach links | ein Zeichen nach links |
| Cursor nach rechts | ein Zeichen nach rechts |
| F1 | WShell-Window auf minimale Größe |
| F2 | WShell-Window auf maximale Größe |
| F5 \ | Wenn man "for" eingibt und anschließend F6 betätigt, wird die History-Liste nach Eingaben, die mit "for" beginnen, (bei F5 nach oben, |
| F6 \ | bei F6 nach unten) durchsucht. |
| F7 | Wort links vom Cursor löschen |
| F8 | Wort rechts vom Cursor löschen |
| F9 | Screen nach vorne/hinten legen |
| F10 | Window nach vorne/hinten legen |

# KICK ME, AMIGA

# Die Software mit dem gewissen KICK




## KICK-ASS

### Der besondere Assembler

KICK-ASS ist im Prinzip mehr als ein Assembler. Durch seinen integrierten EDITOR, MONITOR, DEBUGGER und LINKER ist KICK-ASS ein leistungsstarkes Entwicklungssystem, das die Entwicklung von Assembler-Programmen einfach, bequem und schnell macht. KICK-ASS wurde für all die entwickelt, die viel Wert auf Geschwindigkeit und Komfort legen.
Von den unzähligen Features, die KICK-ASS bietet, möchten wir nur einige nennen:

- Full-Screen-Editor mit vielen komfortablen Befehlen
- blitzschnelle Assemblierung schon während der Eingabe
- Der Quelltext wird stark komprimiert, so daß er sehr wenig Speicherplatz beansprucht.
- automatisches Einrücken der Zeilen, was gerade bei längeren Programmen eine hohe Lesbarkeit bewirkt
- automatische Syntaxüberprüfung schon bei der Eingabe
- Funktionen sind wahlweise durch Pull-Down-Menüs oder über Tastatur aufrufbar.
- Der integrierte LINKER fügt mehrere Programmodule zusammen. Sogar die Verbindung von Assembler mit Kompilaten von höheren Programmiersprachen ist möglich.
- Der DEBUGGER dient der Programmanalyse und Fehlersuche und ist eine unentbehrliche Hilfe beim Hineinschauen in fremde Programme.

**KICK-ASS DM 89.-**

## KICK-ED

### Der universelle Editor

KICK-ED können wir allen, die gern programmieren, sei es in 'C', Assembler, Pascal oder Modula-2, wärmstens empfehlen. Aber auch, 'um 'mal schnell' einen Brief zu schreiben und auszudrucken eignet sich dieser Editor.
KICK-ED ist ein professionelles Werkzeug für Programmierer. Die Bemühung um eine möglichst einleuchtende und angenehme Benutzerführung läßt sich an vielen Details aufzeigen:

- gleichzeitge Verwaltung mehrerer Texte, die mit Hilfe der Funktionstasten angewählt werden können
- Ein Hilfs-Menü, das jederzeit aufgerufen werden kann, gibt vollständige Information über die verschiedenen Befehle.
- bequemes Laden, Speichern und Anhängen von Texten über die integrierte Datei-Auswahl-Box
- Volle Multitasking-Unterstützung des AMIGA.
- superschnelles Softscrolling

**KICK-ED DM 49.-**

### Bestellcoupon

Bitte senden Sie mir:

☐ KICK-ASS DM 89.-
☐ KICK-ED DM 49.-

zuzüglich Versandkosten:
Inland DM 7.50
Ausland DM 10.-

Auslandsbestellungen nur gegen Vorauskasse
Bei Nachnahme zuzüglich DM 3.80 Nachnahmegebühr

Name
Straße
Ort
Datum Unterschrift

MAXON-Computer / Industriestr. 26 / 6236 Eschborn

Auch hier wird wieder deutlich, daß die WShell mehr als ein normales CLI zur Verfügung stellt.

## Muschel mit neuem Innenleben

Die WShell besitzt sogenante "Built-In-Commands". Das bedeutet, daß Kommandos, die man schon aus dem CLI kennt, und einige weitere Befehle nochmals in der WShell realisiert sind. Zu dieser Methode gibt es natürlich geteilte Ansichten. Die Einen sagen, alle Kommandos sollten in der Shell realisiert werden, damit sie nicht immer von Diskette nachgeladen werden müssen (Leute mit mehr als 2MB RAM ...). Die anderen sind der Meinung, daß die mitgelieferten Kommandos genauso gut sind wie die eingebauten und nicht unnütz RAM verbrauchen. In der WShell geht man einen Weg zwischen diesen beiden Polen und hat in nur 2 kB RAM über ein Dutzend Kommandos programmiert. Diese Kommandos sind *cd, echo, else, endcli, endif, failat, goto, if, lab, mounted, pause, popcd, prompt, pushcd, quit, rexx, skip, stack* und *swapcd*. Natürlich können wir hier nicht alle Kommandos erklären, aber die Befehle, die es im CLI (noch) nicht gibt, sollen wenigstens vorgestellt werden.

| | |
|---|---|
| GOTO | Im Gegensatz zu dem schon bekannten Skip-Befehl kann man mit dem GOTO-Befehl an eine beliebige Marke springen. |
| MOUNTED | Mit dem MOUNTED-Befehl kann man feststellen, ob ein bestimmtes Device gemountet ist. Der MOUNTED-Befehl setzt das Fehler-Level auf den Wert 5, falls das Device dem System nicht bekannt ist. |
| PAUSE | entspricht dem ECHO-Befehl, nur wird nach der Textausgabe auf eine beliebige Tastenbetätigung gewartet. |
| POPCD | Der POPCD-Befehl liest vom Directory-Stack den zuletzt dort abgelegten Pfad. |
| PUSHCD | Legt das Directory auf den Stack |
| REXX | Führt ein REXX-Makro aus |
| SWAPCD | Wechsel zwischen dem obersten Directory auf dem Stack und dem aktuellen, wobei das aktuelle Direktory auf dem Stack abgelegt wird. |

Außer diesen sogenannten "Built-In Commands" enthält die WShell-Disk noch einige weitere Programme, die man in das C-Directory seiner Workbench kopieren sollte. Dies sind: *Alias, ConMan, History, LoadLib, NewWSH, PatchWB, Resi, RunWSH, SetCMan, SetExecute* und *SetWSH*. Bis auf den ALIAS-Befehl lassen wir diese Befehle hier unter den Tisch fallen.

Der ALIAS-Befehl ist ein sehr wichtiges Hilfsmittel, um sich viel Tipparbeit in der WShell zu sparen. Betrachten wir uns das an einem Beispiel.
Üblicherweise schreibt man, um eine Diskette zu formatieren, folgende Sequence:

```
FORMAT DRIVE DF0: NAME LEERE_DISK
```

In der WShell schreibt man nur noch das Folgende:

```
FO DF0:
```

Dies ist möglich, nachdem man den folgenden ALIAS-Befehl definiert hat:

```
ALIAS fo  "Format drive [] Name
           Leere_Disk"
```

Hier tritt einer der Schwachpunkte der WShell zutage. Man kann ALIAS-Befehlen leider nur einen Parameter, hier "df0:", übergeben. Die Übergabe des Diskettennamens ist leider nicht möglich. Die Fileflags werden von der WShell besser als vom CLI unterstüzt. Batchdateien mit gesetztem Shell-Flag werden automatisch gestartet, und Programme, bei denen man das E-Flag gelöscht hat, werden nicht mehr ausgeführt.

## Muschel mit Pipeline

Piping?! Auch die Möglichkeit des "Piping" beherrscht die WShell. Piping ermöglicht es, die Ausgaben eines Programms als Eingabe für ein anderes Programm zu verwenden. An einem kleinen Beispiel wollen wir uns dies betrachten. Wir wollen die Ausgabe des DIR-Befehls als Eingabe für ein ARexx-Programm verwenden. Das ARexx-Programm "Tee.rexx" soll seine Eingabe einmal ins CLI und einmal in eine Datei schreiben. Das ARexx-Programm sieht man in einem kleinen Kasten. Auf die genaue Funktion des Programms gehen wir hier nicht ein. Der Aufruf für den beschriebenen Vorgang sieht wie folgt aus:

```
DIR c:  | Tee >InhaltC
```

```
/* Schreibe die Eingaben in ein
              File und ins CLI */
call open stderr,"*"
do until eof(stdin)
   parse pull inline
   say inline
   call writeln stderr,inline
   end
```

*Programm Tee.rexx*

## Auster mit Perle

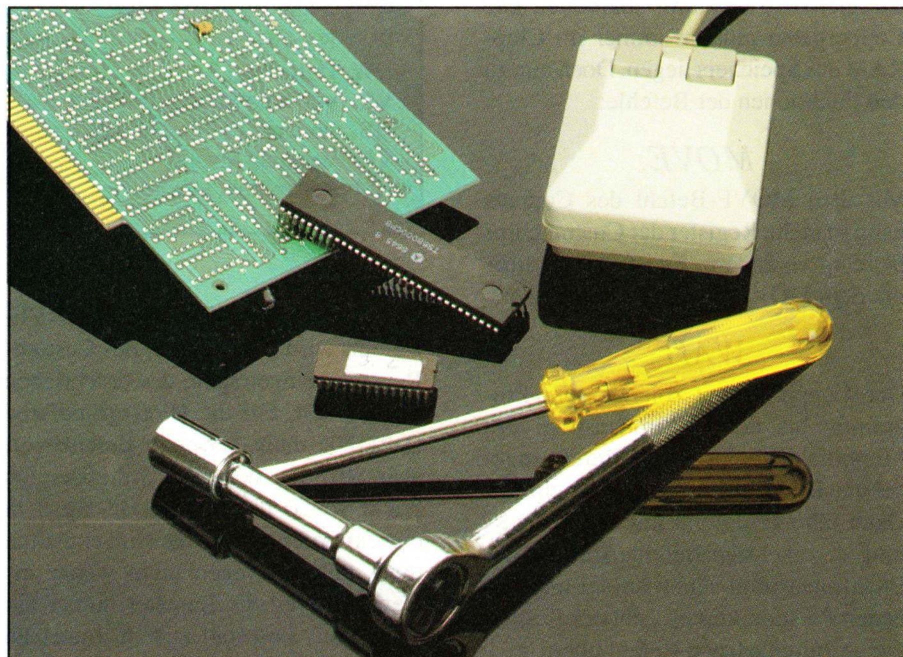
Ich habe die WShell zwei Monate in der Version 1.05 getestet. Zwar habe ich wohl nicht alle Fähigkeiten bis an die Grenze des Möglichen ausgeschöpft, doch war die WShell täglich in einem recht komplexen System im Einsatz. Probleme sind dabei nie aufgetreten. Seit dem es AMIGAs gibt, habe ich so eine Kiste zu Hause stehen, aber noch nie habe ich ein so fehlerloses Anwendungsprogramm in die Finger bekommen. Ob nun DPaint, TxED, CygnusEd, Datamat (hihi), Metacomco Shell oder Compiler, alle haben den einen oder anderen Fehler, nur, mir ist es etwas peinlich, an der WShell habe ich keinen gefunden. Sicher, man könnte einiges besser machen, noch ein paar Funktionen einbauen und so weiter, aber das, was im Handbuch steht (und ich probiert habe), funktioniert mit makelloser Zuverlässigkeit.
Sowie die deutsche Version auf dem Markt ist, kann man jedem, der sich mit dem CLI beschäftigt, die WShell empfehlen. ■

# Die Hardware-programmierung

## Teil 2: DER COPPER

*Nachdem wir Sie im letzten Kurs mit den wichtigsten Grundlagen versorgt haben, greifen wir uns unsere Schnorchel und tauchen etwas tiefer in die Programmierung unserer ach so geliebten Chips ein. Dieser Kursteil beschäftigt sich hauptsächlich mit der Programmierung des Coppers und der Bitplanedarstellung.*

*Von*
*Sven Stillich*
*Oliver Siebenhaar*
*Ralf Görlach*



## Der Copper

Der Copper ist ein sehr interessanter Baustein im AMIGA. Er ist seines Zeichens Coprozessor und übernimmt einfache, aber dennoch sehr wichtige Aufgaben, die den 68000er sehr aufhalten würden. Wie es sich für einen richtigen Prozessor gehört, kann auch der Copper programmiert werden. Er verfügt aber nicht über eine so große Anzahl von Anweisungen wie der Motorola 68000. Er begnügt sich mit nur ganzen drei Befehlen. Diese lauten MOVE, WAIT und SKIP. Der Copper dient hauptsächlich der Grafikprogrammierung. Er erlaubt es nämlich, zu einem festgelegten Zeitpunkt (Bildschirmpunkt/-zeile) ein bestimmtes Customchip-Register zu beschreiben bzw. zu verändern. So ist es zum Beispiel möglich, die Bildschirmfarbe ab einer bestimmten Bildschirmposition zu ändern. Es ergeben sich hieraus ungeahnte Möglichkeiten, angefangen mit fast beliebiger Anzahl von Bildschirmfarben mit nur einer (!) Bitplane, über verschiedene Grafikauflösungen auf einem Bildschirm bis hin zu "schwingenden" Bitplanes. Im "Normalbetrieb" sieht man die Fähigkeiten

des Coppers zum Beispiel dann, wenn man zwei Screens mit verschiedener Auflösung übereinander hin- und herschiebt. Doch nun zur Programmierung.

## Qualität vor Quantität
### die drei Befehle des Coppers

Die Copperbefehle werden hintereinander zu einer sogenannten Copperliste aufgereiht. Eine Copperliste ist somit ein Programm für den Copper und mit einem Assemblerprogramm für den Mikroprozessor zu vergleichen. Dieses Copperprogramm wird immer dann gestartet, wenn der Raststrahl die Nullposition (oben links auf dem Bildschirm) durchläuft. Dadurch ergibt sich, daß der Copper bei jedem Bildaufbau das gleiche Programm abarbeitet. Die Copperlisten müssen immer im Chip-RAM des Speichers liegen. Doch nun zu den Funktionen der Befehle:

### *MOVE:*

Mit dem MOVE-Befehl des Coppers kann man die Register der Customchips beliebig verändern. Er ist vergleichbar mit dem MOVE-Befehl des Prozessors.

### *WAIT:*

Der Copper stoppt solange sein Programm, bis der Elektronenstrahl des Monitors die beim WAIT-Befehl angegebene Bildschirmposition erreicht hat. Dann fährt er normal mit der Bearbeitung der nächsten Befehle fort. Ist der Elektronenstrahl schon an der angegebenen Position vorbei, geht der Copper gleich zum nächsten Befehl über.

### *SKIP:*

Auch beim SKIP-Befehl muß eine Bildschirmposition angegeben werden. Der Copper prüft dann, ob der Elektronenstrahl an dieser Stelle gerade "vorbeistreicht" oder den Punkt schon passiert hat. Ist dies der Fall, dann überspringt er den nächsten Befehl. Ansonsten führt er diesen aus. Der SKIP-Befehl wird allerdings nur selten verwendet.

Alle Befehle bestehen aus zwei Wörtern (16 Bit). Diese zwei Befehlswörter beinhalten die Angaben (Strahlposition, Register), die für den jeweiligen Befehl benötigt werden. Das jeweilige charakteristische Bit ist Bit 0.

| | **MOVE** | | **WAIT** | | **SKIP** | |
|---|---|---|---|---|---|---|
| Bit | Wort1 | Wort2 | Wor1 | Wort2 | Wort1 | Wort2 |
| 15 | - | DW 15 | VS 7 | BFD | VS 7 | BFD |
| 14 | - | DW 14 | VS 6 | VM 6 | VS 6 | VM 6 |
| 13 | - | DW 13 | VS 5 | VM 5 | VS 5 | VM 5 |
| 12 | - | DW 12 | VS 4 | VM 4 | VS 4 | VM 4 |
| 11 | - | DW 11 | VS 3 | VM 3 | VS 3 | VM 3 |
| 10 | - | DW 10 | VS 2 | VM 2 | VS 2 | VM 2 |
| 9 | - | DW 9 | VS 1 | VM 1 | VS 1 | VM 1 |
| 8 | R 8 | DW 8 | VS 0 | VM 0 | VS 0 | VM 0 |
| 7 | R 7 | DW 7 | HS 8 | HM 8 | HS 8 | HM 8 |
| 6 | R 6 | DW 6 | HS 7 | HM 8 | HS 7 | HM 7 |
| 5 | R 5 | DW 5 | HS 6 | HM 8 | HS 6 | HM 6 |
| 4 | R 4 | DW 4 | HS 5 | HM 8 | HS 5 | HM 5 |
| 3 | R 3 | DW 3 | HS 4 | HM 8 | HS 4 | HM 4 |
| 2 | R 2 | DW 2 | HS 3 | HM 8 | HS 3 | HM 3 |
| 1 | R 1 | DW 1 | HS 2 | HM 8 | HS 2 | HM 2 |
| 0 | 0 | DW 0 | 1 | 0 | 1 | 1 |

Erklärung:

| | |
|---|---|
| - | unbenutzt |
| BFD | Blitter Finish Disable |
| R | Register |
| DW | Datenwort |
| VS | vertikale Strahlposition |
| HS | horizontale Strahlposition |
| VM | vertikale Maske |
| HM | horizontale Maske |

In das erste Befehlswort des MOVE-Befehls wird die Nummer des Custom-Registers eingetragen, das beschrieben werden soll. Für die Hintergrundfarbe ($dff180) muß das erste Befehlswort also $0180 lauten. Der Adreßbereich reicht von 0 bis $1BE. Daß Bit 0 in diesem Befehlswort eine Null enthalten muß, fällt deswegen nicht weiter ins Gewicht, weil die Register immer ein Wort breit sind und zudem immer an geraden Adressen liegen (%1=dez.1 kommt also nicht vor). Bit 0 ist bei einer geraden Zahl immer auf Null gesetzt. Das zweite Befehlswort beinhaltet die Daten, die in das jeweilige Register geschrieben werden sollen. Die Berechnung der Datenwörter bei den WAIT- und SKIP-Befehlen gestaltet sich etwas schwieriger, denn die Bits sind etwas verschoben. Beim genaueren Hinsehen erkennt man beim ersten Befehlswort, daß 256 Zeilen abfragbar sind (VS 0 - VS 7). Da der AMIGA aber über 313 Zeilen verfügt, sind die Zeilen 1 ff. und 257 ff. für den Copper gleich. In der Horizontalen sind 112 Positionen zu unterscheiden, allerdings nur in Sprüngen von vier (Bildschirm)-Punkten. Grund dafür ist das Nichtvorhandensein der Bits HS 0 und HS 1. Die zweiten Befehlswörter enthalten die Masken für das erste Befehlswort. Es werden nur diejenigen Bits zur Prüfung benutzt, die auch in den Masken gesetzt sind.

```
WAIT (444,255)     ; Warte, bis der Rasterstrahl die Position x=444
                   ; und y=255 erreicht hat.
WAIT (28,24)       ; Warte, bis der Rasterstrahl die Position x=28
                   ; und y=280 erreicht hat. Denn die Zeilen 0 und
                   ; 256 sind für den Copper gleich. Darum fragt man
                   ; erst die Zeile 255 ab und dann die Restdifferenz.
                   ; Denn 280-256=24
MOVE #$123,$180    ; Verändere dann das Farbregister 0 (Register-Nr $180)
                   ; auf den Wert $123. (R=1,G=2,B=3) =3)
Ende
```

Doch nun ein kleines Beispiel für eine Copperliste. Zur Übung sollten Sie es selbst einmal nachrechnen. Das Programm des Coppers soll folgendermaßen aussehen (siehe vorherige Seite). Die Copperliste kann natürlich weitausmehr Befehle enthalten. Da der Copper ein Programm aber nicht in der Form wie oben gezeigt verstehen kann, müssen wir es noch für ihn übersetzen. Die "Übersetzung" sieht wie folgt aus:

```
dc.w $ffdf,$fffe     ; %1111111111011111,  %1111111111111110
                     ;  765432108765432     B65432108765432
                     ;    VS        HS           VM      HM
                     ; VS=%11111111=255    HS=%110111100=444
dc.w $180f,$fffe     ; %0001100000001111,  %1111111111111110
                     ;  765432108765432     B65432108765432
                     ;    VS        HS           VM      HM
                     ; VS=%00011000=24     HS=%000011100=28
dc.w $0180,$0123     ; %0000000110000000,  %0000000100100011
                     ;  -------87654321     FEDCBA9876543210
                     ;             R           DW
                     ;R=%110000000=$180   DW=%0000000100100011=$123
dc.w $ffff,$fffe     ; zum Abschluß, als Endmarke muß eine unmögliche
                     ; Position angegeben werden. Dies ist eine davon.
```

## Installieren einer neuen Copperliste

Mit dem Erstellen einer eigenen Copperliste ist es nicht getan. Der AMIGA verwendet immer noch die alte Copperliste. Man muß dem Computer mitteilen, wo die neue Copperliste liegt. Dazu gibt es zwei Möglichkeiten. Bei beiden muß zunächst der Copper-DMA ausgeschaltet und nach der Veränderung der Register wieder eingeschaltet werden. Dies geschiet mit dem DMACON-Register.

*Und nun die erste Version:*

```
DMACON  = $DFF096
COP1LC  = $DFF080
COPJMP1 = $DFF088

move.w   #$0080,DMACON        ; Copper-DMA ausschalten
move.l   #Copperlist,COP1LC   ; ab 'Copperlist' beginnt die neue Liste
clr.w    COPJMP1
move.w   #$8080,DMACON        ; Copper-DMA einschalten
```

Hierzu ist zu sagen, daß in das COP1LC-Register die Startadresse der neuen Copperliste geschrieben wird. Daraufhin wird das Register COPJMP1 mit einem beliebigen Wert beschrieben, worauf der Copper auch an dieser Stelle im Speicher beginnt, seine Copperliste abzuarbeiten.

*Die zweite Version lautet:*

```
DMACON = $DFF096
move.l   Gfxbase,a0
move.w   #$0080,DMACON     ; Copper-DMA ausschalten
move.l   50(a0),Oldcopper
move.l   #Copperlist,50(a0)
move.w   #$8080,DMACON     ; Copper-DMA einschalten
```

Ab dem fünfzigsten Byte der graphic. library liegt ein Zeiger auf die Copperliste des Betriebssystems. Der alte Zeiger wird in der dritten Zeile gerettet und in der vierten Zeile auf die neue Copperliste "verbogen". Wenn man wieder die alte Copperliste benutzen möchte, muß man den zuvor geretteten Wert wieder in die vorgesehenen Bytes schreiben.

## Aufbau der Playfields

Nachdem wir nun allerlei Sachen über die Bilddarstellung, verschiedene Auflösungen, DMAs und Copper gehört haben, ist es wohl endlich an der Zeit, einmal mit der Programmierung eines Bildschirms zu beginnen. Doch halt, freuen Sie sich nicht zu früh, denn zuvor sind noch einige Dinge zu klären. Die Bilddarstellung des AMIGA ist an rechteckigen Bereichen orientiert. Es ist also möglich, einen bestimmten rechteckigen Bereich zu definieren, dessen linke obere und rechte untere Ecke man in bestimmte Register schreiben muß, und welcher das Bildschirmfenster darstellt. Der Bereich, der nicht davon eingeschlossen wird, bekommt die Farbe des Registers COLOR00. Die beiden zugehörigen Register heißen DIW STRT ($DFF08E) und DIWSTOP ($DFF090). Sie sind folgendermaßen aufgebaut:

| Bit: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | V7 | V6 | V5 | V4 | V3 | V2 | V1 | V0 | H7 | H6 | H5 | H4 | H3 | H2 | H1 | H0 |

Die Bits H0-H7 enthalten jeweils die x-Koordinate und V0-V7 jeweils die y-Koordinate des Start- (linke obere Ecke des Bildschirmfensters) bzw. Endpunktes (rechte untere Ecke des Bildschirmfensters). Hierbei ist zu beachten, daß DIWSTOP die Endposition + 1 enthalten muß. Soll also das Playfield bis zur Zeile 220 gehen, muß der Wert 221 eingetragen werden. Da die fehlenden Most Significant Bits (MSB) H8 und V8 als 0 angenommen werden, ist die in DIWSTRT festgelegte Startposition auf das linke obere Viertel des Bildschirms beschränkt. Die in DIWSTOP festgelegte HORIZONTALE Endposition ist auf Positionen zwischen 256 und 458 beschränkt, da hier H8 als 1 vorausgesetzt wird. Damit die VERTIKALE Endposition nicht auch einer Begrenzung unterliegt, wird V8 durch Invertieren (Umdrehen) von V7 erreicht. Man kann sich damit vertikaler Endpositionen zwischen 128 und 312 bedienen. Zum besseren Verständnis ein Beispiel: Für Positionen von 128 bis 255 setzt man V7 auf 1, worauf V8 0 wird. Will man eine Endposition zwischen 256 und 312 setzen, so setzt man V7 auf 0 und damit V8 auf 1 (logisch, oder!?).

Das normale Bildschirmfenster des AMIGA hat die DIWSTRT- und DIWSTOP-Werte $2981 (x = 129, y = 41) und $29c1 (x = 449, y = 297) für eine Auflösung von 640 mal 256 bzw. 320

mal 256 Bildpunkten. Die horizontale Austastlücke (der Bereich, in dem nichts dargestellt werden kann, weil der Raststrahl noch unterwegs ist) geht von 30 bis 106, die vertikale von Spalte 0 bis 25. Der sichtbare vertikale Bereich beschränkt sich demnach auf die Zeilen 26 bis 312. Horizontale Startpositionen sind ab 107 möglich.

Nachdem wir nun gelernt haben, wie wir unser Bildschirmfenster definieren müssen, werden wir uns ansehen, wie der Anfang und das Ende der Bitplane-DMAs eingestellt werden. Der Hardware muß nämlich mitgeteilt werden, wann die Daten aus den Bitplanes per DMA gelesen werden sollen, damit diese rechtzeitig auf dem Bildschirm erscheinen. Vertikal beginnt und endet der Bitplane-DMA in den Zeilen, die in DIWSTRT und DIWSTOP angegeben sind. Horizontal wird es schon etwas schwieriger. Um einen Punkt auf dem Bildschirm darzustellen, wird aus jeder Bitplane das aktuelle Wort, in dem sich dieser Punkt befindet, benötigt. Wie wir wissen, können in einem Buszyklus maximal 2 niedrig- oder 4 hochauflösende Punkte dargestellt werden. Daher sind bei 6 Bitplanes in der niedrigen Auflösung 8 und in der hohen Auflösung 4 Buszyklen nötig, um alle Bitplanes zu lesen. Da die Hardware noch einen halben Buszyklus benötigt, um die Daten endlich (!!) auf dem Bildschirm darstellen zu können, muß der Bitplane DMA genau 8.5 (17 niedrig auflösende Punkte) oder 4.5 Buszyklen (9 hochauflösende Punkte) vor dem Anfang des Bildschirmfensters beginnen.

Um dies alles festzulegen, existieren wieder zwei Register: In DDFSTRT (Display Data Fetch Start = $DFF092) wird der Buszyklus des ersten Bitplane-DMAs der Zeile, in DDFSTOP (Display Data Fetch Stop = $DFF094) der des letzten eingetragen. Um es nicht zu kompliziert zu machen, werden wir uns die Erklärung der Formel zur Berechnung der DDFSTRT- und DDFSTOP Werte sparen. Sie hat folgendes Format:

```
DDFSTRT = (HSTART/2 - 8.5) AND $FFF8
DDFSTOP = DDFSTRT + (Anzahl der Punkte pro Zeile /2 - 8)
```

HSTART steht für die horizontale Startposition des Bildschirmfensters. Wollen wir die Datafetch-Werte für das normale Bildschirmfenster des Betriebssystems setzen, so gehen wir wie folgt vor: Im LowRes-Modus hat unser Fenster 320 Punkte pro Zeile, der horizontale Start des Bildschirmfensters befindet sich bei $81. Setzen wir diese Werte in die Formel ein, so bekommen wir folgendes Ergebnis:

```
DDFSTRT = ($81/2 - 8.5) AND $FFF8 = $38
DDFSTOP = $38 + (320/2 - 8)       = $d0
```

Im HiRes-Modus lautet die Formel anders:

```
DDFSTRT = (HSTART/2 - 4.5) AND $FFFC
DDFSTOP = DDFSTRT + (Anzahl der Punkte pro Zeile /4 - 8)
```

Beim Setzen der Datafetch-Werte ist darauf zu achten, daß DDFSTRT nicht kleiner als $18 und DDFSTOP nicht größer als $D8 ist. Bei DDFSTRT-Positionen kleiner als $34 überlappen sich die DMA-Zyklen der Bitplanes und Sprites, was zur Folge hat, daß einige Sprites nicht mehr darstellbar sind. Wie bei George Orwells 1984 geht es auch beim AMIGA nicht ohne Kontrolle (CPU is watching you!). Am Hebel der Macht sitzen bei der Bilddarstellung die drei Register BPLCON0, BPLCON1 und BPLCON2, die unter anderem bestimmen, wieviele Bitplanes aktiv sind, welcher Darstellungsmodus aktiviert ist, und die ferner für Scrolling und Prioritäten der Playfields zuständig sind. Um uns mit diesen doch sehr mächtigen Registern gutzustellen und sie nicht durch unqualifizierte Programmierung zu verärgern, müssen wir ihren Aufbau genau kennen (siehe Tabelle BPLCON0 ($DFF 100).

Interessant sind für uns besonders die Bits 12-14. Sie bilden gemeinsam zwar keine starke Gemeinschaft, jedoch immerhin eine 3Bit Zahl, mit der wir angeben können, wieviele Bitplanes uns genehm sind. Die besonders aufmerksamen Leser (es soll sie ja wirklich geben!) werden sich (oder uns!) jetzt sicherlich fragen, wie es möglich ist, daß man mit einer 3Bit-Zahl zwar Werte bis einschließlich sieben darstellen kann, der AMIGA aber bekanntlicherweise nur höchstens sechs Bitplanes aktivieren kann. Nun hier wird einfach auf Ihr Geschick vertraut, die Anzahl der aktiven Planes nicht den magischen Wert SECHS überschreiten zu lassen. Nur Mut, mit ein bißchen Übung kriegen Sie das schon hin. Wenn nicht, ist es auch nicht schlimm, es werden dann einfach nur sechs Planes dargestellt. Doch genug des Palavers, kommen wir zu einem konkreten Beispiel: Wir wollen im LoRes-Modus eine Grafik mit 32 Farben darstellen. Es sind also fünf Bitplanes erforderlich. Die 3Bit-Zahl heißt also %101. Zusätzlich setzen wir noch Bit 9, weil wir unsere 32 Farben ja nicht nur als Graustufen bewundern wollen. Der Wert, den wir also in das Register BPLCON0 schreiben müßten, wäre:

BPLCON0 ($DFF100):

| Bit Nr. | Name | Funktion |
|---|---|---|
| 15 | HIRES | Wenn 1, dann HiRes-Modus |
| 14 | BPU2 | 3Bit-Zahl, die die Anzahl |
| 13 | BPU1 | der aktivierten Bitplanes |
| 12 | BPU0 | angibt (0-6). |
| 11 | HOMOD | Wenn 1, dann HAM-Modus |
| 10 | DBPLF | Wenn 1, dann Dual-Playfield |
| 9 | COLOR | Wenn 1, dann Videoausgang Farbe |
| 8 | GAUD | Wenn 1, dann Genlock Audio an |
| 7-4 | — | nicht benutzt |
| 3 | LPEN | Wenn 1, dann Lightpen-Eingang an |
| 2 | LACE | Wenn 1, dann Interlace-Modus an |
| 1 | ERSY | Wenn 1, dann externe Synchronisation an |
| 0 | —— | nicht benutzt |

```
%0101001000000000.
```

Die übrigen Modi werden wir in späteren Beispielen genauer behandeln (Immer mit der Ruhe!). Der nächste Kontrolleur ist nicht der blonde Hans von der Bundesbahn, sondern das Register BPLCON1, das es uns erlaubt, die geraden (2,4 und 6) und die ungeraden (1,3 und 5) Bitplanes unabhängig voneinander um bis zu 15 Bildpunkte nach rechts zu verschieben. Durch geschickte Programmierung läßt sich damit ein 'budderweisches Sgrolling' erreichen. Dazu muß man jedoch zwei zusätzliche Register zu Rate ziehen. Es sind sogenannte Modulo-Register, die auch bei der Blitter-Programmierung (kommt alles noch) eine wichtige Rolle spielen. Sie heißen BPL1MOD ($DFF108) für die ungeraden und BPL2MOD ($DFF10A) für die geraden Planes. Man muß sich das folgendermaßen vorstellen: Es ist wie bei einem Kameramann, der ein Fußballspiel aufzeichnet. Er sieht durch die Kamera immer nur einen Teil des Spielfeldes, obwohl dieses ganz vorhanden ist. Genauso ist es beim AMIGA. Die Kamera ist der Ausschnitt, der gerade auf dem Monitor sichtbar ist, das Spielfeld ein beliebig großes, im Speicher liegendes Grafikfeld. Wie bei dem Kameramann, der vom linken Spielfeldrand zum rechten schwenkt, weil dort wieder mal eine Prügelei im Gange ist, läßt sich beim AMIGA der sichtbare Ausschnitt über das große Spielfeld bewegen. Doch wie geht das nun im einzelnen vor sich? Nun, angenommen, wir hätten ein 80 Byte (40 Worte) breites Grafikfeld im Speicher, während das sichtbare Bildschirmfenster nur 40 Byte (20 Worte) breit ist. Bei der Bilddarstellung würden jetzt, da das Grafikfeld zusammenhängend im Speicher liegt, nach Darstellung der ersten 40 Byte der ersten Zeile, die nächsten 40 Byte dieser Zeile, die ja eigentlich nicht sichtbar sein sollte, als zweite Zeile des Bildschirmfensters dargestellt. Das verhindern die Modulo-Register. Nach Darstellung der ersten Zeile der eingestellten Bildschirmbreite (in unserem Fall 40 Byte), wird der Wert der jeweiligen Modulo-Register zu BPLxPT addiert und so die angegebene Bytezahl dieser Zeile übersprungen. In unserem Beispiel müßten dies 40 Byte sein, so daß nur die ersten 40 Byte jeder Zeile sichtbar sind. Um nun die zweite Hälfte unseres Grafikfeldes "vorbeiscrollen" zu lassen, sehen wir uns zunächst das

```
        ; HardwareKurs (II) Beispielprogramm - Copperlist/Cycle...
        ; 1988,89 Sven Stillich, Oliver Siebenhaar (KICKSTART)
        ; Ausgabe Juni 1989
        ;definieren der CUSTOMchip Reg. und der
        ;Library Offsets (Bases)

allocmem       = -198          ; speicher reservieren
freemem        = -210          ; speicher freigeben
openlibrary    = -408          ; library öffnen
execbase       =  004          ; basisadresse EXEC
        ; Blitter- und Bitplaneadressen (CUSTOMCHIPBASE = $DFF000)
COP1LC         = $080          ; adresse copperlist
COPJMP1        = $088          ; sprung copperlist
DIWSTRT        = $08E          ; Fenster START
DIWSTOP        = $090          ; Fenster STOP
DDFSTRT        = $092          ; BitplaneDMA START
DDFSTOP        = $094          ; BitplaneDMA STOP
DMACON         = $096          ; DMA control
BPL1PTH        = $0E0          ; bitplane1 basis high
BPL1PTL        = $0E2          ; bitplane1 basis low
BPLCON0        = $100          ; bitplanecontrol 0
BPLCON1        = $102          ; control 1 (scroll)
BPLCON2        = $104          ; control 2 (prior)
BPL1MOD        = $108          ; ungerade modulo
BPL2MOD        = $10A          ; gerade modulo
SPR0PTH        = $120          ; sprite0 basis high
SPR0PTL        = $122          ; sprite0 basis low
COLOR00        = $180          ; hintergrundfarbe
CIAAPRA        = $BFE001       ; CIAportregister A
chip           = $02           ; parameter für den
clear          = chip+$10000   ; 'allocmem'-aufruf
startlist      = 38            ; basis alte cliste
               ;
               ; Programmbeginn
               ; Speicher reservieren (für bitplane/copperlist)
               ;

    LEA     $DFF000,a5             ; custombase nach a5
               MOVE.L  execbase.s,a6
               MOVE.L  #[40*290],d0            ; [320/8]*290 Bytes
               MOVE.L  #clear,d1               ; bedingungen
               JSR     allocmem(a6)            ; speicher bitplane
               MOVE.L  d0,bitplanebase
               MOVE.L  #156,d0                 ; 39 * 4byte Befehle
               MOVEQ   #chip,d1
               JSR     allocmem(a6)            ; speicher copperlist
               MOVE.L  d0,copperbase
               ;
          ; Berechnung der Copperliste (Sprite/BitplanePointer)
               ;
               MOVE.L  d0,a0                   ; a0 - copperbase
               MOVE.L  bitplanebase(pc),d0     ; d0 - bitplanebase
               MOVE.L  #spritebase,d1          ; d1 - spritebase
               MOVE.W  #BPL1PTH,(a0)
               MOVE.W  #BPL1PTL,4(a0)
               MOVE.W  #SPR0PTH,8(a0)
               MOVE.W  #SPR0PTL,12(a0)
               MOVE.W  d0,6(a0)                ; bpllow
               SWAP    d0
               MOVE.W  d0,2(a0)                ; bplhigh
               MOVE.W  d1,14(a0)               ; sprlow
               SWAP    d1
               MOVE.W  d1,10(a0)               ; sprhigh
               MOVE.W  #COLOR00,16(a0)
               MOVE.W  #$0000,18(a0)           ; background schwarz
               ;
               ; Colorspreading berechnen ab Zeile (d0)
               ;
               ADD.L   #20,a0                 ; Parameter überspringen
               MOVEQ   #00,d0
               MOVE.W  #$500F,d0               ; ab Zeile $50
               LEA     colortabelle(pc),a1     ; farbwerte
               MOVEQ   #15,d1                  ; 16 farbwerte
copper_set:    MOVE.W  d0,(a0)+                ; WAIT
```

→

BPLCON1 ($DFF102):

| Bit Nr.: | 15-8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Funktion: | unbel. | P2H3 | P2H2 | P2H1 | P2H0 | P1H3 | P1H2 | P1H1 | P1H0 |

```
                MOVE.W  #$FFFE,(a0)+
                MOVE.W  #COLOR00,(a0)+      ; MOVE
                MOVE.W  (a1)+,(a0)+
                ADD.W   #$0100,d0
                DBF     d1,copper_set
                MOVE.W  #COLOR00,(a0)+      ; background wieder
                MOVE.W  #$0000,(a0)+        ; schwarz
                MOVE.L  #$FFFFFFFE,(a0)     ; copperlist ende
                ;
                ; Copperlist initialisieren / Parameter setzen
                ;
                MOVE.W  #$03E0,DMACON(a5)             ; DMA sperren
                MOVE.L  copperbase(pc),COP1LC(a5)     ; basis setzen
                CLR.W   COPJMP1(a5)                   ; liste starten
                MOVE.W  #$2081,DIWSTRT(a5)
                MOVE.W  #$2FC1,DIWSTOP(a5)
                MOVE.W  #$0038,DDFSTRT(a5)
                MOVE.W  #$00D0,DDFSTOP(a5)
                MOVE.W  #%0001001000000000,BPLCON0(a5)
                ;        5432109876543210
                CLR.W   BPLCON1(a5)
                CLR.W   BPLCON2(a5)
                CLR.W   BPL1MOD(a5)
                CLR.W   BPL2MOD(a5)
                MOVE.W  #$83C0,DMACON(a5)
                ;
                ; ColorCyclen im durch WAIT/MOVE-Befehle
                ; definierten Bereich
cycle:          MOVE.L  copperbase(pc),a0
                ADD.L   #26,a0          ; auf ersten Befehl zeigen
                LEA     colortabelle(pc),a1 ; colortabelle
                ADDQ.L  #02,tabulator       ; tab + 2
                CMP.L   #31*2,tabulator     ; farben am ENDE?
                BLE.S   do_it
                CLR.L   tabulator           ; ja => zurücksetzen
do_it:          ADD.L   tabulator(pc),a1    ; tab hinzuaddieren
                MOVEQ   #15,d0              ; 16 Zeilen
set:            MOVE.W  (a1)+,(a0)          ; neuer wert in clist
                ADDQ.L  #08,a0              ; nächster MOVE-Befehl
                DBF     d0,set              ; ende der Schleife
                MOVE.L  #20000,d0           ; Verzögerung
wait:           DBF     d0,wait
mouseklick:     BTST    #06,CIAAPRA
                BNE.S   cycle
                ;
                ; Programmende
                ; (alte Copperlist zulassen/speicher zurück)
                LEA     gfxname(pc),a1      ; graphics.library
                JSR     openlibrary(a6)     ; öffnen, um basis der
                MOVE.L  d0,a6               ; alten clist zu finden
                MOVE.L  startlist(a6),COP1LC(a5)  ; basis alte clist
                CLR.W   COPJMP1(a5)         ; alte clist starten
                MOVE.W  #$8020,DMACON(a5)   ; dma zurücksetzen
                MOVE.L  execbase.s,a6
                MOVE.L  bitplanebase(pc),a1 ; basis bitplane
                MOVE.L  #[40*290],d0        ; size bitplane
                JSR     freemem(a6)         ; speicher freigeben
                MOVE.L  copperbase(pc),a1   ; basis copperlist
                MOVE.L  #156,d0             ; size copperlist
                JSR     freemem(a6)         ; speicher freigeben
                MOVEQ   #00,d0
                RTS
                ;
                ; Variablen/Parameter/Tabellen/Strings
                ;
                EVEN
copperbase:     DC.L 0
bitplanebase:   DC.L 0
tabulator:      DC.L -2
spritebase:     DC.W 0,0
colortabelle:   DC.W $ F00,$0F10,$0F20,$0F30,$0F40,$0F50,$0F60
                DC.W $0F70,$0F80,$0F90,$0FA0,$0FB0,$0FC0,$0FD0
                DC.W $0FE0,$0FF0,$0FE0,$0FD0,$0FC0,$0FB0,$0FA0
                DC.W $0F90,$0F80,$0F70,$0F60,$0F50,$0F40,$0F30
                DC.W $0F20,$0F10,$0F00
                ;          kopie der ersten farbwerte
                DC.W $0F00,$0F10,$0F20,$0F30,$0F40,$0F50,$0F60
                DC.W $0F70,$0F80,$0F90,$0FA0,$0FB0,$0FC0,$0FD0
                DC.W $0FE0,$0FF0,$0FE0,$0FD0,$0FC0,$0FB0,$0FA0
                DC.W $0F90,$0F80,$0F70,$0F60,$0F50,$0F40,$0F30
                DC.W $0F20,$0F10,$0F00
gfxname:        DC.B 'graphics.library',0
```

Register BPLCON1 genauer an. Jeweils 4 Bits bilden einen Wert (höchstens 15), der angibt, um wieviel Pixel der Bidschirm nach rechts verschoben werden soll. P1H0-P1H3 betreffen dabei die geraden, P2H0-P2H3 die ungeraden Planes. Um nun ein Scrolling unseres Grafikfeldes von rechts nach links zu erreichen, zählt man den (die) 4Bit-Wert(e) in BPLCON1 von 15 schrittweise bis 0 herunter und erhöht dann BPLXPT um 1 Wort, da ja 16 Bit nach links verschoben wurden. Nun geht es wieder von vorne los: Runterzählen und anschließend 1 Wort addieren. Man muß nur darauf achten, daß rechts noch Bilddaten vorhanden sind, sonst kommt auf einmal irgendein unbekannter Speicherbereich vorbeigerauscht.

Umgekehrt (von links nach rechts) geht es genauso, das heißt: Raufzählen und anschließend 1 Wort subtrahieren.

Lassen Sie sich von diesem doch ein wenig komplizierten Vorgang nicht die Lust an unserem interessanten (gähn) Kurs verderben. Wir werden später alles mit Beispielen belegen.

Der Dritte im Bunde der Mächtigen ist das Register BPLCON2 ($DFF104), das über die Prioritäten der Sprites in bezug auf die Playfields und der Playfields untereinander entscheidet. Da uns das jetzt jedoch materiell rein gar nichts bringt, werden wir dieses Register ausführlich bei der Programmierung von Sprites und Playfieldspielereien behandeln.

*Doch nun zu unserem Programm:*

## DAS PROGRAMM (Copperliste)

Aller guten Dinge sind drei, und aller Anfang eines Assemblerprogrammes besteht aus Definitionen von Konstanten usw. Bei uns sind es dieses Mal die Offsets für die Aufrufe der Betriebssystemroutinen "allocmem", "freemem" und "openlibrary" (klar) sowie die Basisadresse der exec.library (glasklar). Danach folgen die Customchipregister. Diese haben wir ja schon weiter oben erklärt. Die jeweiligen Funktionen der Register sind aber im Quellcode kurz beschrieben. Da wir ja flexibel sind, nehmen wir nicht den Speicherbereich,

der uns paßt, sondern lassen ihn uns mittels "allocmem" vom System zuweisen. Nur so können wir sicher sein, daß wir auch wirklich CHIP-Mem für adressieren. Es ist zwar ein bißchen mehr als wir brauchen (290 Zeilen), aber wenn wir unser Programm später noch weiterverwenden wollen (Scrolling, ...), dürfen wir nicht mit Speicher geizen. Die Anfangsadresse des Speichers wird in "bitplanebase" zwischengespeichert. Nun wird dem System noch der Speicher für die interne Copperliste entrissen ("copperbase").

Jetzt geht's los. Also aufpassen (oder "achtpassen", wie unser Informatiklehrer immer sagt)! Ab Zeile 55 beginnt nämlich die Berechnung der Copperliste. Sie beginnt mit drei MOVE-Befehlen, die gleich zum Anfang die Startadresse der Bitplane festlegen, den Mauszeiger (SPRITE00) löschen sowie die Hintergrundfarbe unseres neugeborenen Bildschirms auf Schwarz setzen. Hierzu wurde eine geniale Routine ersonnen, die der Meister (nicht Du, Thorsten!) sogleich erklären wird. So höret und staunet: Grundlegenderdings wird die Anfangsadresse unserer Bitplane durch ein Langwort dargestellt. Liegt unsere Plane also zum Beispiel ab $72718 im Speicher, lautet die Startadresse $00072718. Unser Copper kann jedoch nur Words lesen. Die Lösung dieses Problems liegt in den beiden Wörtern "Hoch" (HIGH) und "Tief" (LOW). Was haben diese uns höchstens aus dem Wetterbericht bekannten Begriffe mit dem Copper zu tun? Nun, nehmen wir unsere Bitplaneadresse. Sie besteht aus zwei Words, einem HÖHERwertigen ($0007) und einem NIEDRIGERwertigen ($2718). Diese teilen wir nun den Registern BPL1PTH und BPL1PTL mit und - fertig. Unsere Routine macht nun folgendes: Sie MOVEt zuerst das LOWword der Startadresse hinter BPL1PTL (Zeile 63). Dann wird das Register D0 geSWAPt, das heißt, die Words werden vertauscht, so daß D0 dann den Inhalt $27180007 hat(Zeile 64), und MOVEt dann das Word $0007 hinter BPL1PTH (Zeile 65). Das gleiche geschieht mit der neuen Anfangsadresse des Mauspointers. Wie das mit den Sprites richtig funktioniert, folgt in späteren Kursteilen. Im Moment ist nur wichtig, daß der Zeiger weg ist. Unsere Copperlist besteht in Zeile 71

```
MOVE #bitplanebaseHIGH    ,BPL1PTH  (Bytes 00-03)
MOVE #bitplanebaseLOW     ,BPL1PTL  (Bytes 04-07)
MOVE #spritebaseHIGH      ,SPR0PTH  (Bytes 08-11)
MOVE #spritebaseLOW       ,SPR0PTL  (Bytes 12-15)
MOVE #$0000 (schwarz)     ,COLOR00  (Bytes 16-19)
```

also aus den Befehlen:
Ab Zeile 74 folgt die Berechnung des Rests der Copperliste. Zuerst einmal wird zu A0 (Basisadresse der Copperliste) 20 hinzuaddiert, um die schon berechneten Befehle zu überspringen. Als erste Zeile unseres Farbbalkens haben wir $50 gewählt, Ihr könnt aber ruhig damit experimentieren. Das Wort ($500f) in D0 stellt schon das erste Befehlswort eines WAIT-Befehls dar, so daß man später nur noch $0100 dazuzählen muß, um eine Zeile weiter zu kommen ($510f). Nun laden wir die Basisadresse der 16 Farbdaten in A1. In der nun folgenden Schleife werden immer ein WAIT- mit folgendem MOVE-Befehl generiert. Das heißt, der Copper wartet auf eine Bildschirmposition, setzt dann die Farbe in der nächsten Zeile setzt er eine andere Farbe, usw. Auf diese Weise ist es möglich, alle Farben des AMIGA zu verwenden, ohne auch nur eine Bitplane angeschaltet zu haben. Die folgenden Befehle unserer

```
WAIT auf Zeile $50          (Bytes 20-23)
MOVE #$0f00,COLOR00         (Bytes 24-27)
WAIT auf Zeile $51          (Bytes 28-31)
MOVE #$0f10,COLOR00         (Bytes 32-35)
```

Liste lauten also:
Am Ende der Copperliste wird die Bildschirmfarbe wieder auf Schwarz gesetzt (Zeile 86). Das sieht einfach schöner aus. Ihr könnt ja die beiden Zeilen (86/87) mal weglassen. So, am Ende der ganzen Copperliste setzen wir LISTig, wie wir sind, einen unmögliche WAIT-Befehl, den der Copper als Endmarkierung auffaßt und wieder am Anfang der Liste anfängt. Als nächstes folgt die Initialisierung der neuen Copperliste und einige Einstellungen der Bildschirmdarstellung, auf die wir jetzt nicht weiter eingehen wollen (oben erklärt).

## DAS PROGRAMM (ColorCyclen)

So, jetzt wird's schwierig, aber - Null Problemo. Wir kommen zum Herzstück unseres kleinen Programms, dem ColorCyclen. Dieser Effekt ist wohl allen AMIGA-Benutzern bekannt, die schon einmal mit DELUXE PAINT herumexperimentiert haben. Hierbei werden die Inhalte der Farbregister rotiert. Alles ganz simpel. In der Praxis bedeutet dies, daß die ... äh, ja ... nun die Inhalte werden halt ... hierzu ein Beispiel:

```
Urzustand:
COLOR00 - Inhalt: $0f00
COLOR01 - Inhalt: $0123
COLOR02 - Inhalt: $0afb

erste Rotation:
COLOR00 - Inhalt: $0afb
COLOR01 - Inhalt: $0f00
COLOR02 - Inhalt: $0123

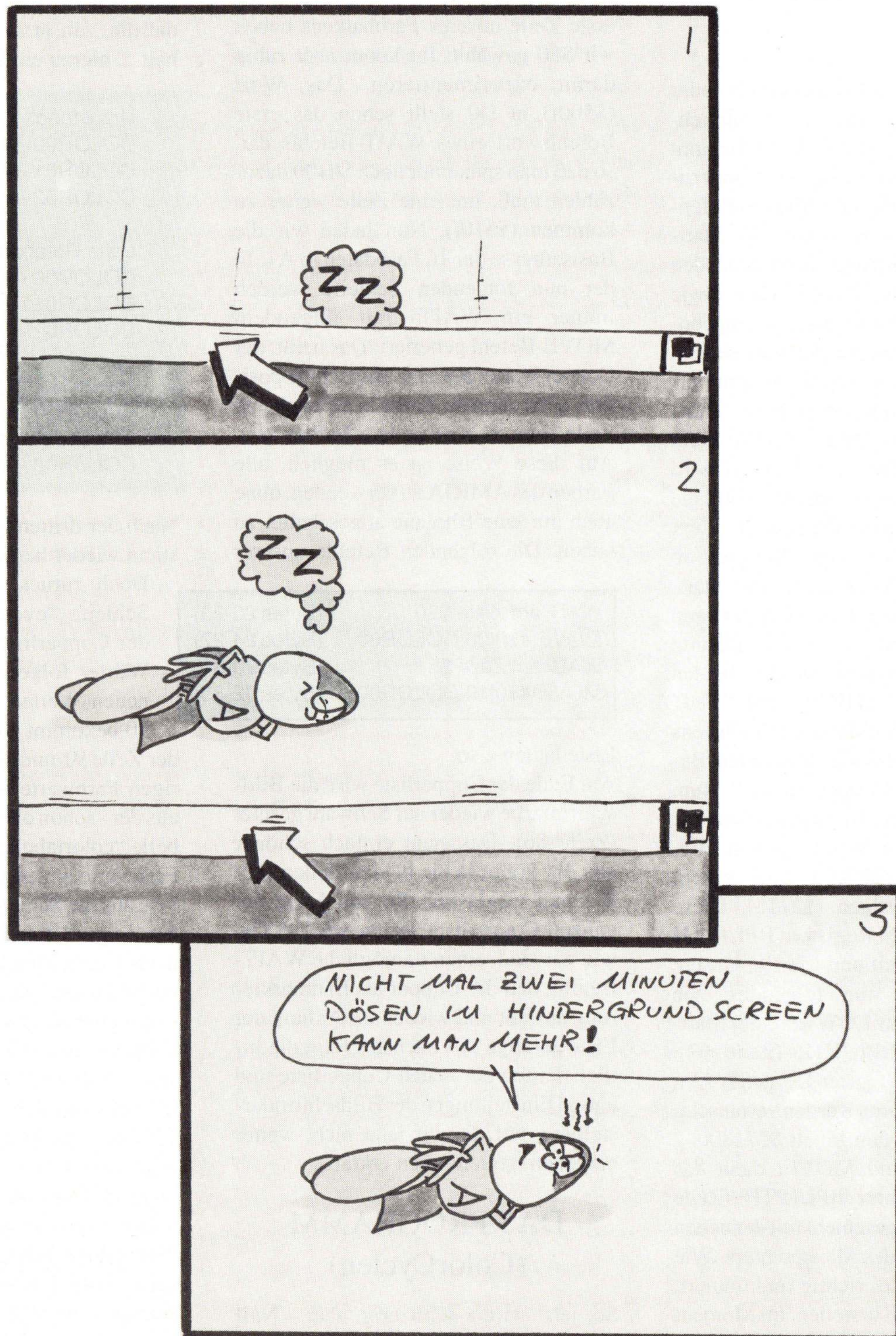
zweite Rotation:
COLOR00 - Inhalt: $0123
COLOR01 - Inhalt: $0afb
COLOR02 - Inhalt: $0f00
```

Nach der dritten Rotation ist der Urzustand wieder hergestellt.

Doch zurück zum Programm. Die Schleife "cycle:" beschreibt die in der Copperliste auf die COLOR00-Wörter folgenden Datenwörter mit neuen Werten. Die Bildschirmzeile 90 bekommt zum Beispiel den Wert der Zeile 91 und so weiter. Die zugehörigen Farbwerte bekommt die Routine aus der - schon oben erwähnten - Farbtabelle "colortabelle" (ab Zeile 154). Die Farbwerte liegen zweimal vor. Warum? Wie aus der Schleife ersichtlich ist, wird ein Zeiger ("tabulator") in der Tabelle nach jedem Durchlauf immer um eins vorgeschoben. Dieser Zeiger zeigt (was soll er auch sonst machen) immer auf die Farbe der ersten Zeile (in unserem Beispiel: Zeile $50). Die nachfolgenden 15 Bildschirmzeilen bekommen die dem Tabulator nachfolgenden 15 Farbwerte zugewiesen. Nun wandert unser Zeiger (aha, er kann also nicht bloß zeigen) immer weiter in unserer Tabelle voran. Was nun passieren würde, wenn wir keine zwei Farbtabellen hätten, kann man sich am besten klarmachen, wenn man sich zwei DIN A4-Blätter auf einem schwarzen Tisch vorstellt.

Die schmale Seite des einen Blattes sind unsere 16 Copperzeilen, die des anderen unsere 6 Farbwerte. Nun legen wir die beiden Blätter deckungsgleich übereinander, das heißt, die erste Zeile hat die Farbe des ersten Werts in der Tabelle. So schön, so gut. Nun erhöhen wir den "Zeiger", das heißt, wir verschieben das eine Blatt nach unten. Was passiert? Für die letzten Copperbalken sind keine Farbwerte mehr vorhanden. Tragisch, tragisch. Des Rätsels Lösung sind - wer hätte es geahnt - ZWEI Farbtabellen. Ich hoffe, Sie haben's kapiert, dann war die Mühe nicht umsonst (aber der Untertitel dieses Kursteils ist jetzt wenigstens gerechtfertigt). Zum Schluß: Nach der Mausabfrage wird dem System der allocierte Speicher zurückgegeben, die alte Copperliste wieder installiert und das Programm verlassen.

*So, das war's mal wieder. Wir sparen uns eine Vorschau, um Sie nicht zu sehr zu verschrecken. Als Überbrückung haben Sie an diesem Kursteil ja noch genug zu knabbern. Tschüß.*




PH 9/89 © MAXON

VON LARS SCHMITT

# Modem Talking

*Diesmal geht es nach Frankfurt, wo Commodore eine Mailbox betreibt. Sie nennt sich Combo und steht im Hause Commodore!*

## COMBO

### Hält Combo, was Commodore verspricht?

Doch neben den allgemeinen Mailbox-Diensten bietet die Combo kaum herausragende Neuerungen. Wirklich vom Sockel gehauen hat mich z.B. der Menüpunkt "Uhrzeit anzeigen", denn bis die atomuhrgenaue Uhrzeit beim Empfänger ankommt, da ist schon wieder 5 Minuten später als vor drei Tagen ... Toll dagegen sind zweifelsohne die Anruferzahlen. Die Combo erfreut sich regen Zuspruchs, so daß man Tag und Nacht nur sehr schlecht Zugang zu ihr findet (gelobt sei die Wahlwiederholung). Dementsprechend sieht es auch in den öffentlichen Pinboards aus. Da kann man sehr oft Nachrichten lesen, die nur ein paar Stunden alt sind. Positiv ist auch die Aufteilung des Pinboards in eigene (Computer-)Rubriken. So kommt man am schnellsten an die gewünschten Informationen ran. Unter Menüpunkt DREI ist auch an den AMIGA gedacht. Dagegen kann man über Menüpunkt 9 des Hauptmenüs nur leicht grinsen. Hat man den Fehler gemacht, und eine UN-WIEDER-GUT-ZU-MACHENDE "9" eingegeben, so wird das Drama zunächst durch eine Ladezeit eingeleitet (Eliza ist extern gespeichert). Hat sich Eliza erst ordnungsgemäß gemeldet, dann beginnt der zweite Teil des Gemetzels. Schließlich folgt einer der aufregendsten Flirts, den ich je mit Bits und Bytes führte. Die wirklich Oskar-verdächtige Unterhaltung ist im Protokoll der Nachwelt überliefert (Auch zu finden im Köchel-Verzeichnis unter Ablage "T" wie "Trashcan"!). Hat man endlich das Wort "BYE" getippt, kündigt sich der dritte und letzte Akt mit den Worten "PLEASE WAIT!" an, denn das Mailboxsystem muß wieder geladen werden (und Commodore-Disks sind ja sooooooo schnell!).

Wirklich gut sollte der Menüpunkt 10 (FREETIME in Frankfurt) werden. Da dreht sich wenigstens mal was um das Frankfurter Kultur- und Nachtleben. Doch die 14 Monate alten "CLUB-CHARTs" erinnerten mich stark an das Durchblättern einer antiquarischen Pop-Enzyklopädie. Die vom Verfasser zum Teil selbst getesteten Lokale mögen zwar für den Frankfurter recht interessant sein..... Naja, der Bayer (oder alle anderen Nicht-Frankfurter) muß halt Menüpunkt 10 umgehen (am besten weiträumig, wie die ewige Baustelle auf der A66!). Schade für mich, wo ich doch jeden Tag beruflich in Frankfurt bin, aber diverse Tips sind leider zu alt.

Leider muß ich zum Schluß noch denjenigen enttäuschen, der denkt, über die Combo an aktuelle Informationen von Commodore zu gelangen. Daß da ein aktueller Testbericht über einen "neuen" Commodore-64-Emulator" für den AMIGA ins Hauptmenü übernommen wurde, zeugt von der atemberaubenden Geschwindigkeit, mit der der Commodore-Mailbox-Operator die BOX überarbeitet.

Lesen Sie sich in aller Ruhe das Protokoll durch, denn schließlich müssen Sie als Anwender entscheiden, ob die Combo viel wert ist oder nicht. Für mich bietet sie allerdings nichts Neues. Die interessante Box "FREETIME" ist leider nicht "up to date" und für auswärtige Anrufer wohl auch uninteressant. Eliza war der zweite, allerdings auch schon der letzte, Gag der Box. Schon viele "kleine" Mailboxen bieten da mehr Pep und Witz in der Aufmachung. Nicht zu übersehen ist jedoch das riesige Angebot an aktuellen Pinboardeinträgen. Das Pinboard der Combo ist wirklich zu empfehlen. Also, wer was zu verkaufen hat, oder wer was sucht, der sollte man bei Combo aktiv werden. Allerdings: Wer hinter Combo ein Super-Mailboxsystem der Firma Commodore erwartet hat, das zudem noch aktuell mit Commodore gekoppelt ist, der hat sich geirrt - LEIDER.

Tröstlich ist allerdings, daß ein Eintrag als Benutzer kostenlos läuft. Der Eintrag zum Empfangen von Mails läuft sogar schon beim ersten Login automatisch ab (wodurch sich auch die hohe Benutzerzahl erklären läßt).

```
==================================
            COMBO
      Die Mailbox der
 Commodore Bueromaschinen GmbH Ffm
==================================

V39 - Sie sind der 118727.  Anrufer!

Benutzernr.? (0 = Erklaerung)? 3758
Passwort?
Mal sehen, ob ich Sie kenne....

Guten Tag, Lurchi aus Hofheim!
Sie benutzten COMBO zuletzt
am 10.04.1989 um 05:28:51 Uhr

=> timeout = 60 Sekunden !
-------------------
mon/10.apr.1989/053642
>>>>> COMBO HAUPTMENUE <<<<< (t=000044)

    1 = Kurzanleitung COMBO
    2 = Uhrzeit anzeigen
    3 = Informations Ecke
    4 = Liste der COMBO Benutzer
    5 = Dialog mit dem SysOp
    6 = Pinboard ansehen
    7 = Pinboard Info eingeben
    8 = Mailbox-Service
    9 = Dialog mit ELIZA
   10 = FREETIME in Frankfurt
   11 = Technische Anfragen
   12 = Betreff: Mitarbeit an COMBO
   13 = Neuheit: Amiga emuliert C-64
   99 = zurueck zum Hauptmenue
ende = Verbindung abbauen

       C O M B O    Kurz-Anleitung:

beim Anruf wird als erstes Zeichen  ein
RETURN verlangt.
Wenn kein Zeichen mehr kommt,  schaltet
COMBO nach 60 sec. automatisch ab.

CTRL S (ASCII 19)  haelt  eine  Ausgabe an.
CTRL Q (ASCII 17) Fortsetzung der
                  Ausgabe
CTRL C (ASCII 3) oder
CTRL X (ASCII 24) Abbrechen der Ausgabe
                  Rueckkehr zum Menue

RETURN            Menue wird komplett
                  nochmal angezeigt.

Eine ausfuehrliche Anleitung finden Sie
in der Informations Ecke.

Ihre COMBO

>>>>> COMBO HAUPTMENUE <<<<< (t=000338)
(1 - 13,99,?,ende)? 10
 >>>>Allgemeines zu FREETIME<<<<

 "FREETIME" ist die Box in Combo, die
 sich ganz bewusst nicht mit comput-
 erspezifischen Informationen befasst
 sondern mit Informationen aus dem
 Frankfurter Raum bezueglich, des
 Nachtlebens und verschiedenen, an-
 deren Arten der Freizeitbeschaefti-
 gung im Bereich von Ffm. Diese Box
 ist momentan noch in der Entstehungs-
 phase, fuer Anregungen und konstruk-
 tiver Kritik, in die MB 3 0 0 0, sei
 Euch im voraus bereits gedankt. Nun
 in kurzer Uebersicht die bis jetzt
 geplanten Programmpunkte:

1.CINEMA (Berichte ueber Filme)
2.Frankfurt Inside (Klatsch&Infos)
3.Who's who in Francfort? ?
4.Club Charts (Hitparade aus Ffm)
5.Adressen (von Bistro's,Cafe's,Bar's,Diskotheken etc.)
6.Highlight's (Termine)
7.Plattenvorstellungen
8.Fashion
9.Konzerte/Schauspiel/Oper/Ballet

VIEL SPASS................MARK

>>>> PINBOARD SUB NEU <<<<< (t=000928)

    1 = Commodore 64
    2 = PC und kompatible
    3 = Amiga und 68000er
    4 = Andere Rechner
    5 = Computer Zubehoer
    6 = Software
    7 = Sonstiges (keine Computer)
    8 = Kontakte (Gruesze)
    9 = Mailboxnews
   10 = Mails an User ohne Mb
   11 = Diskussions - Board
   99 = zurueck zum Hauptmenue
ende = Verbindung abbauen

 COMBO hat z.Z. 6664 Benutzer!

(1 - 13,99,?,ende)? 9

einen Moment...

hi Lurchi, i'm eliza. please tell me
your problem. type 'bye' to leave me.
? you are my problem, eliza what makes
you think i am your problem, eliza?
? you are mad say, do you have any
psychological problems?
? no why no?
? yes are you sure?
? yes please don't repeat yourself.
? yes please don't repeat yourself.
? no don't say no it's always so negative.
? thats the reason, why i say it
what answer would please you the most?
? go home what does that suggest to you?
? bye

good bye, Lurchi.

please wait!

(1 - 11,99,?,ende)? ende

Sie benutzten COMBO von
sam/01.apr.1989/171701 bis sam/01.apr.1989/174551

Sie haben in 28 Min.
30535 Bytes empfangen und somit 59% Auslastung erzielt!

Auf Wiedersehen, Lurchi aus Hofheim.

Trennen Sie nun die Verbindung.
```

## COMBO Mailbox-System

### *Steckbrief:*

*Mailbox:* 069-6638191 24h300bdfd 8/N/1
*Programm:* kommerziell (durch einen Combo-User geschrieben, 25 KByte)
*Hardware:* Zentraleinheit 8296SK mit Speeddos / 8010 IEEE 488-Modem DCF-77 Funkuhr-Empfänger
*POSITIV:* haufenweise Anrufe / viele aktuelle Pinboardeinträge Board "FREETIME"
*NEGATIV:* keine Commodore-News / keine Besonderheiten
*GESAMT:* befriedigend

VON ANDREAS BIERNAT

# Die Floppy im Griff !!!

## Teil 4:

## Die Programmierung des Diskettencontrollers

*In diesem Teil geht es um die direkte Programmierung des Disk-Controllers. Wir wollen erst einmal langsam anfangen, und uns ein wenig mit dem Motor befassen. Bevor wir Daten lesen oder schreiben können, müssen wir uns mit der Codierung und Decodierung der Daten befassen. Es gibt also noch viel zu tun.*



Listing 7 demonstriert die Benutzung des Motors. Damit wir bei der Laufwerksprogrammierung nicht vom Multitasking gestört werden (Zugriffe auf die Laufwerke sind nie auszuschließen), wird es kurzerhand mit der Funktion FORBID der Exec-Library gesperrt. Falls es in extremen Fällen dazu kommen sollte, daß sogar die Interrupts unser Programm stören sollten, dann benutzen wir die Funktion DISABLE anstelle von FORBID, welche ebenfalls das Taskswitching sperrt und dazu noch alle Interrupts verbietet. Die Interrupts sollte man aber nicht zu lange verbieten, da sie für Exec notwendig sind, Vorsicht ist geboten.

Als zweiter Teil der Initialisierung wird die Disk-DMA erlaubt, indem das entsprechende Bit im DMACON-Register gesetzt wird. Nun bringt man das Laufwerk in Gang. Dazu wird als erstes das Select-Bit gesetzt. Danach wird das Motor-Bit aktiviert (auf Null gesetzt) und das Select-Bit gelöscht. Wenn also die Select-Leitung zum Laufwerk einen Sprung von 1 auf 0 durchführt, merkt sich das Laufwerk den Zustand des Motor-Bits, wobei 0 Motor an und 1 Motor aus bedeutet. Danach wird der Schreib-/Lesekopf auf Spur Null gebracht. Wir wissen ja beim Aufruf des Programms nicht, auf welchem Zylinder er sich gerade befindet. Falls er auf Spur Null ist, ist Bit 4 des Registers $BFE001 gelöscht. Ist es das nicht, wird der Kopf um eine Spur nach außen gefahren. Die Bewegung des Kopfes wird über einen 0-1-Sprung des Bits 0 im Register $BFD100 ausgelöst, während die Richtung durch Bit 1 bestimmt ist (1 nach außen, 0 nach innen).

Zwischen zwei Motorbewegungen muß eine gewisse Zeit gewartet werden, denn der Motor ist nicht so schnell wieder ansprechbar. Diese Motorbewegung wird so oft wiederholt, bis der Kopf auf Spur Null angelangt ist. Dabei zählen wir mit, wieviel Tracks wir bis dahin übersprungen haben und erhalten so die

ursprüngliche Trackposition. Der Kopf muß nämlich am Ende des Programms wieder auf seine ursprüngliche Position gefahren werden, da sonst die Position, die sich das Betriebssystem gemerkt hat, nicht mehr stimmt. Nachdem der Kopf nun auf Spur Null steht, wird er über die ganze Diskette auf Spur 79 bewegt. Je nach dem Wert der Warteschleife (delay), geht das schneller oder langsamer und hört sich auch anders an. Mit dem Delay-Wert sollte man mal etwas experimentieren. Allerdings klappt bei delay = 0 gar nichts mehr!
Um das Programm nun ordnungsgemäß zu verlassen, wird zuerst der Kopf auf die ursprüngliche Position gebracht. Dann wird der Laufwerksmotor ausgeschaltet und das Laufwerk deselektiert. Nun kann auch das Taskswitching wieder zugelassen werden (und eventuell gesperrte Interrupts wieder freigegeben werden). Dies geschieht mit den Funktionen PERMIT oder ENABLE der Exec-Library.

Noch ein Wort zum Programmierstil. Wenn man sich nun seine eigenen Routinen schreibt, sollte man dies ordentlich mit Unter- bzw. Einzelroutinen machen. Das Demonstrationslisting ist nämlich nur "von oben nach unten" hingeschrieben, ohne übersichtliche Unterroutinen. Wer sich ein wenig genauer mit den Libraries auskennt, kann seine eigenen Routinen auch zu einer Library zusammenfassen. Als Beispiel sei eine Unterroutine zur Bewegung des Schreib-/Lesekopfes zu erwähnen. Die Parameter wären dann nur die Nummer des Laufwerks und die Nummer des Zylinders, zu dem der Kopf gebracht werden soll. Den Rest sollte dann die Routine erledigen können.

## Die Codierung

Kommen wir nun zur Codierung der Daten. Die benutzten Routinen sind alle dem Betriebssystem entnommen. Da es mittlerweile viele Versionen des Betriebssystems gibt, habe ich die Sprungadressen variabel gamacht, so daß man die Routinen problemlos in seine eigenen Programme einbinden kann. Die erste Frage, die wir uns zu stellen haben, ist, wie wir die Codierung ins MFM-Format programmtechnisch realisieren können.
Die Antwort auf diese Frage ist nicht leicht zu verstehen, man muß sich schon intensiv damit beschäftigen. Es handelt sich hierbei um mehrere logische Operationen (AND, OR, EOR). Sehen wir uns zuerst eine Routine an, die einen Block-Header codiert. Der Block-Header besteht aus einem Langwort (Formatkennung, Tracknummer, Sektornummer, Entfernung zur Lücke). Dieses Langwort wird in D0 übergeben, während in A0 ein Zeiger auf den Speicherbereich übergeben wird, in den der codierte Header gespeichert werden soll. Die Routine heißt "codlongword" und ist in Listing 8 zu finden. Um diesen Algorithmus zu verstehen, wollen wir ihn mit einem Beispiel nachvollziehen. Sei unser Header $FF150802. Wir gehen nun alle Schritte einzeln durch.

```
$FF150802 =>
  %1111 1111 0001 0101 0000 1000 0000 0010
```

Diese Bits werden nach rechts verschoben, da zuerst die ungeraden Bits codiert werden:

```
  %0111 1111 1000 1010 1000 0100 0000 0001
```

Nun werden alle ungeraden Bits (Taktbits) gelöscht:

```
  %0111 1111 1000 1010 1000 0100 0000 0001
AND
  %0101 0101 0101 0101 0101 0101 0101 0101
  %0101 0101 0000 0000 0000 0100 0000 0001
```

Als nächstes werden alle geraden Bits (Datenbits) umgedreht:

```
  %0101 0101 0000 0000 0000 0100 0000 0001
EOR
  %0101 0101 0101 0101 0101 0101 0101 0101
  %0000 0000 0101 0101 0101 0001 0101 0100
```

Nun werden alle umgedrehten Datenbits einmal nach rechts und einmal nach links verschoben. Diese beiden Ergebnisse werden miteinander UND-Verknüpft. Außerdem wird das vorderste Bit des nach rechts geschobenen Langworts gesetzt.

```
  %1000 0000 0010 1010 1010 1000 1010 1010
AND
  %0000 0000 1010 1010 1010 0010 1010 1000
  %0000 0000 0010 1010 1010 0000 1010 1000
```

In dem so erhaltenen Langwort existieren nur noch 1-Bits, wenn in dem ursprünglichen Langwort neben den jetzigen 1-Bits zwei Nullbits stehen. Die gesetzten Taktbits sind also schon richtig eingefügt. Wenn nun dieses Langwort mit dem Langwort, in dem alle Taktbits entfernt wurden, verknüpft wird, haben wir den fertig codierten Wert:

```
  %0101 0101 0000 0000 0000 0100 0000 0001
OR
  %0000 0000 0010 1010 1010 0000 1010 1000
  %0101 0101 0010 1010 1010 0100 1010 1001
```

Nun muß noch das erste Datenbit des vorherigen Wertes geprüft werden. Ist es gesetzt, muß das letzte Taktbit unseres Langwortes gelöscht werden. Ist es gelöscht, wird das letzte Datenbit (Bit 30) unseres Langwortes geprüft. Ist es gesetzt, so ist alles o.k., sonst wird das letzte Taktbit (Bit 31) unseres Langwortes gesetzt. Das erste codierte Langwort ist also: $552AA4A9. Analog geht es mit den geraden Bits:

```
  %1111 1111 0001 0101 0000 1000 0000 0010
AND
  %0101 0101 0101 0101 0101 0101 0101 0101
  %0101 0101 0001 0101 0000 0000 0000 0000
EOR
  %0101 0101 0101 0101 0101 0101 0101 0101
  %0000 0000 0100 0000 0101 0101 0101 0101
  %1000 0000 1000 0000 1010 1010 1010 1010
AND
  %0000 0000 0010 0000 0010 1010 1010 1010
  %0000 0000 0000 0000 0010 1010 1010 1010
OR
  %0101 0101 0001 0101 0000 0000 0000 0000
  %0101 0101 0001 0101 0010 1010 1010 1010
```

Da das erste Datenbit des vorherigen Langwortes (Bit 0) gesetzt ist, muß das letzte Taktbit des zweiten Langwortes (Bit 31) gelöscht werden. Da es schon gelöscht ist, ist die Codierung beendet. Der codierte Header lautet also:

```
$552AA4A9 55152AAA
```

Diese Form der Codierung ist recht aufwendig und langsam. Deshalb hat man sich überlegt, ob nicht der Blitter diese Arbeit übernehmen könnte. Die Routinen der Blitter-Codierung sind recht lang; ich werde sie später besprechen, da sie zu diesem Zeitpunkt nur für Verwirrung sorgen würden. Es ist schon viel verlangt, das obige Codierverfahren zu versichern. In der gleichen Weise, wie wir das Codieren gelernt haben, wollen wir das Decodieren lernen. Hier also zuerst eine Routine, die einen Block-Header decodiert. Aus zwei aufeinanderfolgenden Langworten wird ein uncodiertes Langwort erstellt. A0 zeigt auf den Speicherbereich, in dem das erste Langwort steht. Als Ergebnis steht das decodierte Langwort in D0.

→

decodiereheader:

```
move.l (a0)+,d0          ;erstes Langwort holen
move.l (a0)+,d1          ;zweites Langwort holen
and.l #$55555555,d0      ;Taktbits entfernen
and.l #$55555555,d1      ;Taktbits entfernen
lsl.l #1,d0              ;Bits richtigstellen
or.l  d1,d0              ;und verknüpfen
rts                      ;fertig
```

Das sieht doch schon viel einfacher aus. Auch hier werden wir den Algorithmus anhand eines Beispiels nachvollziehen. Unsere codierten Langwörter waren $552AA4A9 und $55152AAA. Herauskommen müßte $FF150802.

```
$552AA4A9 =>
  %0101 0101 0010 1010 1010 0100 1010 1001
$55152AAA =>
  %0101 0101 0001 0101 0010 1010 1010 1010
```

Zuerst werden die Taktbits (alle ungeraden Bits) entfernt:

```
  %0101 0101 0010 1010 1010 0100 1010 1001
AND
  %0101 0101 0101 0101 0101 0101 0101 0101
  %0101 0101 0000 0000 0000 0100 0000 0001
  %0101 0101 0001 0101 0010 1010 1010 1010
AND
  %0101 0101 0101 0101 0101 0101 0101 0101
  %0101 0101 0001 0101 0000 0000 0000 0000
```

Nun wird das Langwort, welches die ungeraden Bits darstellt (das erste Langwort) um eins nach links geschoben und mit dem zweiten OR-verknüpft:

```
  % 1010 1010 0000 0000 0000 1000 0000 0010
OR
  % 0101 0101 0001 0101 0000 0000 0000 0000
  % 1111 1111 0001 0101 0000 1000 0000 0010
  => $FF150802
```

Wie man sieht, kommt der ursprüngliche Block-Header wieder heraus. Die Decodierung ist zwar wesentlich schneller als die Codierung, doch das ist kein Grund, sich nicht auch Blitter-Routinen für die Decodierung auszudenken. Zum Verständnis der Blitter-Routinen für die Codierung und Decodierung, muß noch etwas geklärt werden. Um den exklusiven Zugriff auf den Blitter zu bekommen, wird die Funktion QBLIT aufgerufen. Dieser Funktion wird ein Zeiger auf eine Struktur übergeben, die an das Ende einer Liste solcher Strukturen gehängt wird. Der Blitter arbeitet die Strukturen dieser Liste der Reihe nach ab. Wenn eine Struktur an der Reihe ist, hat sie alleinigen Zugriff auf den Blitter, ohne vom Multitasking gestört zu werden. Die Funktion QBLIT wartet also bis der Blitter frei ist, um dann die Kontrolle über ihn an ein Programm weiterzugeben, welches in der Struktur festgelegt ist. Die Struktur sieht folgendermaßen aus:

| Offset | Länge | Beschreibung |
|---|---|---|
| 00 | dc.l | Zeiger auf nächste Struktur |
| 04 | dc.l | Zeiger auf auszuführendes Programm |
| 08 | dc.l | Länge der Daten zur Codierung |
| 12 | dc.l | Zeiger auf Quelle |
| 16 | dc.l | Zeiger auf Ziel |
| 20 | dc.w | Inhalt für BLTSIZE |
| 22 | dc.l | (Von der Anwendung abhängiger Wert) |
| 26 | dc.l | Zeiger auf Laufwerks-Port |

Den Zeiger auf die nächste Struktur setzt die QBLIT-Funktion; er braucht uns also nicht zu interessieren. Der Zeiger auf das auszuführende Programm muß auf die eigene Routine zeigen, die angesprungen wird, wenn der Blitter frei ist. Den Zeiger auf den Laufwerks-Port erhält man aus der Device-Struktur mit dem Offset 36.

```
lea IOStdReq,a0
move.l 20(a0),a0
move.l 36(a0),a0 => Laufwerks-Port für df0: in A0
```

Die Zeiger für die weiteren Laufwerke stehen direkt dahinter:

```
df1: 40
df2: 44
df3: 48
```

Wenn die QBLIT-Funktion unser Programm aufgerufen hat, ist die Funktion noch nicht abgeschlossen. Nur wenn unser Programm in D0 eine Null zurückgibt, erfolgt ein Rücksprung aus der QBLIT-Funktion.

Die Routine zum Codieren eines gesamten Blocks und ein Demonstration der Benutzung dieser Routine stellt Listing 8 dar. Die gesamten Operationen finden im Speicher statt, es wird noch nicht mit dem Laufwerk gearbeitet. Die Quelle stellt den Inhalt des Datenblocks dar. Dieser wird zuerst mit lauter Nullen initialisiert. Dann wird das eigene Programm in den Bereich kopiert, damit wir nicht bloß Nullen codieren.

Das Trackdisk-Device wird hier einmal anders geöffnet. Genau wie die Open-Device-Funktion die Liste der Devices durchsucht, suchen wir nach unserem Device. Danach wird ein kleiner Trick angewandt. Da normalerweise der Trackdisk-Task die Routinen aufruft, bekommt er auch die Messages über seinen Reply-Port. Wenn nun unser Task auf die Messages wartet, werden diese an den Trackdisk-Task gesendet und unser Task könnte warten, bis der Strom ausfällt. Deshalb muß in der Reply-Port-Struktur der Empfänger-Task unser eigener Task sein anstatt des Trackdisk-Tasks. Wir tragen also einen Zeiger auf unseren Task in das mp_SigTask-Feld ein und erhalten somit die Messages. Am Ende des Programms wird dieser Zeiger wieder richtiggestellt.

Während der Arbeit mit den Routinen darf uns der Trackdisk-Task nicht stören, deshalb muß er in einen Zustand gesetzt werden, in dem er keine weiteren Arbeiten mehr übernimmt. Dazu existiert ein Bit im Laufwerks-Port bei Offset 34 (unit_flags). Ist das Bit 0 gesetzt, ist das Device in Arbeit und kann keine weiteren Aufgaben erledigen.

Das Programm codiert zuerst den Block-Header. Man könnte sich hier fragen, warum die zu codierenden Nullbytes nicht direkt als $AAAA-Worte in den Zielpuffer geschrieben werden, sondern den Umweg über die Codier-Routine gehen. Hier muß aber beachtet werden, daß bei der Codierung der vorherige Wert für den gerade bearbeiteten ausschlaggebend sein kann.

Dieses Problem löst die Routine "connect". Wenn die codierten Nullbytes nicht richtig mit den vorherigen Werten verkettet werden, kann es zu Fehlern bei der Decodierung kommen. Die fünf Blitter-Routinen sind recht schwer zu verstehen. Die Wahl der Minterms ist ausschlaggebend für die Art der Verknüpfung. Für genaueres Verständnis muß man schon ein Fachmann im Umgang mit dem Blitter sein. Die Routine zum Decodieren eines einzelnen Langwortes haben wir ja schon kennengelernt. Die Tatsache, daß wir es nur mit zwei verschiedene Verknüpfungen zu tun haben, zeigt sich in der Kürze der Blitter-Routinen. Bei der Codierung mußte der Blitter viermal aufgerufen werden, während nun alles in einem Durchgang erledigt wird. Betrachten wir also Listing 9.

Dieses Programm ist alleine nicht ausführbar, es enthält nur die Routinen zur Decodierung. Fügt man diese in Listing 8 ein, erhält man ein vollständig lauffähiges Programm. Man muß beim Aufruf der Decodierungsroutine aber beachten, daß die Quelle der Decodierung das Ziel+64 der Codierung ist, da sonst die Daten des Block-Headers mit decodiert werden. Bei der Decodierung der

Synchronisationsmarke kommt nämlich ein unbrauchbarer Wert heraus, da sich der Synchronisationswert nicht durch normale Codierung erreichen läßt. Soll also das Ergebnis der Decodierung dasselbe wie vor der Codierung sein, muß die Decodierungsroutine mit obenbeschriebener Quelle aufgerufen werden. Wenn aber das Block-Header-Langwort decodiert werden soll, ruft man einfach die Routine "decodelongword" auf, die wir schon kennengelernt haben. Es wird, wie bei der Codierung, nur im Speicher decodiert; es finden noch keine Diskettenzugriffe statt. Die Initialisierung der Trackdisk-Devices ist ebenfalls die gleiche, denn wir haben es mit denselben Problemen zu tun. Im nächsten und letzten Teil kommen wir dann zu den Schreib- und Leseroutinen.

```

 ; Listing 7

 execbase = 4
 disable = -120
 enable = -126
 delay = 3000

 run:
  move.l execbase,a6   ;Exec-Basisadresse
  jsr    disable(a6)   ;Interrupts (+ Multitasking)
                        sperren
  move.w #$8210,$dff096  ;Disk-DMA erlauben
  bset   #3,$bfd100      ;Laufwerk deselektieren
  bclr   #7,$bfd100      ;Motor-Bit aktivieren
  bclr   #3,$bfd100      ;Laufwerk selektieren =>
                          Motor ein

 ;Kopf nach Zylinder 0 fahren

  bset   #1,$bfd100    ;fahre nach aussen
  clr.l  d2            ;Register für alten Zylinder
 seek:
  move.w #delay,d1       ;Wert für Warteschleife
 seekdelay:
  dbra   d1,seekdelay    ;Warteschleife
  btst   #4,$bfe001      ;Kopf auf Zylinder 0?
  beq    seekend         ;Ja! Dann verzweige
  bclr   #0,$bfd100      ;Kopf einem Schritt bewegen
  add.l  #1,d2           ;nächster Zylinder
  bset   #0,$bfd100      ;Nur 1-0 übergänge werden
                          registriert
  bra    seek            ;Solange, bis Kopf auf Cyl 0
 seekend:
  move.l d2,oldcyl       ;alten Zylinder speichern

 ;Kopf nach Zylinder 79 bewegen

  move.l #78,d7;Aktueller Zylinder momentan Cyl 79
 loop:
  bclr   #1,$bfd100 ;fahre nach innen
 markel:
  move.w #delay,d1  ;Wert für Warteschleife
 marke2:
  dbra   d1,marke2  ;Warteschleife
  bclr   #0,$bfd100 ;Kopf einen Schritt bewegen
  bset   #0,$bfd100 ;Nur 1-0 übergänge registriert
  dbra   d7,markel  ;Bis Kopf auf Zylinder 79 ist

 ;Kopf auf alte Position bringen

  move.l #79,d7            ;aktueller Zylinder
  move.l oldcyl,d6         ;alter Zylinder
  bset   #1,$bfd100 ;fahre nach aussen
 alt:
  move.w #delay,d1  ;Wert für Warteschleife
 warte:
  dbra   d1,warte   ;Warteschleife
  cmp.l  d6,d7      ;alten Zylinder erreicht?
  beq    ende       ;Ja, dann verzweige
  bclr   #0,$bfd100 ;Kopf bewegen
  sub.l  #1,d7      ;nächster Zylinder
  bset   #0,$bfd100 ;Nur 1-0 Übergänge werden
                     registriert
  bra    alt        ;Bis alter Zylinder erreicht
 ende:
  bset   #3,$bfd100 ;Laufwerk deselekt. (für 1-0
                     Übergang)
  bset   #7,$bfd100 ;Motor
  bclr   #3,$bfd100 ;ausschalten
  bset   #3,$bfd100 ;Laufwerk deselektieren
  jsr    enable(a6) ;Interrupts und Multitasking
                     erlauben
 rts                ;Programmende

 oldcyl: dc.l 0

```

*Listing 7*

```
 ;Listing 8
 execbase = 4
 putmsg = -366
 getmsg = -372
 findname = -276
 wait = -318
 qblit = -276
 header = $ff150802

 run:
  lea    quelle,a0   ;Zeiger auf Quelle
  lea    run,a1      ;Zeiger auf eigenes Programm
  move.l #127,d0     ;128 Langwörter kopieren
 copyloop:
  move.l (a1)+,(a0)+ ;Programm als Quelle
                      (Inhalt des Datenblocks)
  dbf    d0,copyloop ;insgesamt 512 Bytes
  move.l execbase,a6 ;Zeiger auf Execbase
  lea    trdname,a1  ;Zeiger auf Device-Name
  lea    350(a6),a0  ;Zeiger auf Device-List
  jsr    findname(a6)  ;Device finden
  tst.l  d0            ;gefunden?
  beq    notfound      ;Nein, ende
  move.l 276(a6),a0    ;Zeiger auf eigenen Task
  move.l d0,a6         ;Device-Struktur in A6
  move.l 36(a6),a3     ;Zeiger auf Laufwerk-Port
                        (df0:)
  lea    174(a3),a1    ;Adresse des Reply-Ports
  move.l 16(a1),-(a7)  ;Zeiger auf Empfänger-Task
                        retten
  move.l a1,-(a7)      ;Reply-Port retten
  move.l a0,16(a1)     ;eigenen Task als Empf.
                        eintragen
  bset   #0,34(a3)     ;Trackdisktask auf Wartepos.
                        setzen
  move.l #header,d0    ;Block-Header in D0
  lea    quelle,a0     ;Zeiger auf Quelle in A0
  lea    ziel,a1       ;Zeiger auf Ziel in A1
  jsr    blockcodierung ;Block codieren
  move.l (a7)+,a1       ;Replyport zurückholen
  move.l (a7)+,16(a1)   ;alten Empfänger-Task
                         wieder setzen
  bclr   #0,34(a3)      ;Trackdisk-Task wieder
                         freigeben
 notfound:
  rts                   ;Fertig

 blockcodierung:
 ;A0 Zeiger auf zu codierenden Block (Quelle)
 ;A1 Zeiger auf Puffer für codierte Daten (Ziel)
 ;A3 Zeiger auf Message-Port des Lauferks
 ;A6 Zeiger auf Trackdisk-Device-Struktur
 ;D0 enthält den uncodierten Informationsteil des
                                     Block-Headers
  movem.l a2/a4/d2,-(a7) ;Register retten
  move.l a1,a4 ;Zeiger auf Ziel nach A4 kopieren
  move.l a0,a2 ;Zeiger auf Quelle nach A2 kop.
  move.l d0,d2 ;Block-Header nach D2 kopieren
  moveq  #0,d0         ;D0 löschen
```

→

```
    lea      (a4),a0          ;A0 zeigt auf Quelle
    bsr      codlongword      ;Nullbytes codieren -->
                              in Puffer
    move.l   #$44894489,4(a4) ;Sync-Markierungen
                              eintragen
    move.l   d2,d0            ;Block-Header wieder nach D0
    lea      8(a4),a0         ;Zeiger auf Quelle
    bsr      codlongword      ;Block-Header codieren
                              und eintragen
    moveq    #3,d2            ;Zähler auf 3 setzen
nullcod:
    moveq    #0,d0            ;Nullbytes
    bsr      codlongword      ;4 Nullbytes codieren und
                              eintragen
    dbf      d2,nullcod       ;Das Ganze 4 mal => 16
                              Nullbytes
    lea      8(a4),a0         ;A0 zeigt auf codierten
                              Block-Header
    moveq    #40,d1           ;40 Bytes für Prüfsumme
    bsr      psumme           ;Blockprüfsumme berechnen
    lea      48(a4),a0        ;Position der Prüfsumme
    bsr      codlongword      ;Prüfsumme codieren und
                              eintragen
    move.l   #512,d0          ;Anzahl Bytes im Datenblock
    move.l   a2,a0            ;Zeiger auf Datenpuffer
    lea      64(a4),a1        ;Zeiger im Zielpuffer setzen
    bsr      coddatablock     ;Datenblock codieren
    lea      64(a4),a1        ;Zeiger auf Anfang der
                              cod. Daten
    move.w   #1024,d1         ;1024 Bytes für Prüfsumme
    bsr      psumme ;Prüfsumme der Daten berechnen
    lea      56(a4),a0        ;Position der Prüfsumme
    bsr      codlongword ;Prüfsumme codieren und
                          eintragen
    movem.l  (a7)+,a4/a2/d2   ;Register zurückholen
    rts                       ;Fertig
codlongword:
;D0 enthält das zu codierende Langwort
;A0 zeigt auf Speicherb.,in den die zwei codier.
;Langwörter hineingeschrieben werden.
    movem.l  d2-d3,-(a7)      ;D2 und D3 retten
    move.l   d0,d3            ;Langwort nach D3 kopieren
    lsr.l    #1,d0 ;Einmal nach rechts verschieben
    bsr      codiere ;ungerade Bits codieren und
                      eintragen
    move.l   d3,d0 ;Langwort wieder in D0
    bsr      codiere ;gerade Bits codieren und
                      eintragen
    bsr      connect ;Taktbit nächsten Wertes
                      berichtigen
    movem.l  (a7)+,d2-d3      ;Register zurückholen
    rts                       ;Fertig
codiere:
;Routine wird selbstständig aufgerufen und
normalerweise nicht
;direkt vom Programmierer benutzt.
;D0 enthält das zu codierende Langwort
;A0 zeigt auf Adresse, an der das cod. Langwort
geschrieben wird
    and.l    #$55555555,d0    ;ungerade Bits
                              (Taktbits) entfernen
    move.l   d0,d2            ;Ergebnis zwischenspeichern
    eor.l    #$55555555,d2    ;gerade Bits (Datenbits)
                              umdrehen
    move.l   d2,d1            ;Ergebnis nach D1
    lsl.l    #1,d2 ;Einmal nach links verschieben
    lsr.l    #1,d1 ;Einmal nach rechts verschieben
    bset     #31,d1 ;vorderstes Bit setzen
    and.l    d2,d1 ;Verknüpfen, Taktbits elimi-
                    nieren
    or.l     d1,d0 ;Taktbits setzen
    btst     #0,-1(a0);Ergebnis Nullbit am Anfang?
    beq      bitok ;Ja, Bits stehen richtig
    bclr     #31,d0 ;sonst vorderstes Bit löschen
bitok:
    move.l   d0,(a0)+ ;Wert eintragen
    rts               ;Rücksprung
connect:
;Routine wird selbstständig aufgerufen und
normalerweise nicht
;direkt vom Programmierer benutzt.
;A0 zeigt auf das nachfolgende Byte
    move.b   (a0),d0          ;Byte holen
    btst     #0,-1(a0) ;erstes Datenbit (Bit 0) des
                       ;vorherigen Wertes gesetzt?
    bne      deletebit ;Ja, dann Taktbit löschen
    btst     #6,d0     ;letztes Datenbit (Bit 30)
                       prüfen
    bne      bitsok    ;wenn gesetzt, dann alles OK
    bset     #7,d0     ;letztes Taktbit (Bit 31)
                       setzen
    bra      eintragen ;korrigierten Wert zurück-
                       schreiben
deletebit:
    bclr     #7,d0     ;Taktbit löschen
eintragen:
    move.b   d0,(a0)   ;Wert zurückschreiben
bitsok:
    rts                ;Rücksprung
coddatablock:
;A0 zeigt auf den Datenpuffer (Quelle)
;A1 zeigt auf den Zielpuffer
;A3 zeigt auf den Laufwerks-Port
;A6 zeigt auf die Device-Struktur
;D0 enthält die Anzahl Daten, die codiert werden
sollen
    link     a2,#-30 ;Platz für QBLIT-Struktur.
                      A2 zeigt
                     ;nun auf das Ende der Struktur
    move.w   d0,d1 ;Anzahl Daten nach D1 kopieren
    lsl.w    #2,d1 ;durch 4 teilen
    ori.w    #8,d1 ;BLTSIZE-Registerinhaltbestim.
    move.w   d1,-10(a2) ;und in Struktur eintragen
    movem.l  a1/a0/d0,-22(a2) ;Datenl., Quelle und
                               Ziel eintragen
    move.l   #qbroutine,-26(a2) ;Zeiger auf auszu-
                                 führendes Programm
    move.l   a3,-4(a2) ;Zeiger auf Laufwerks-Port
                        eintragen
    lea      -30(a2),a1 ;A1 zeigt auf Anfang der
                         Struktur
    move.l   a6,-(a7)     ;A6 speichern
    move.l   56(a6),a6    ;Zeiger auf GfxBase in A6
    jsr      qblit(a6)    ;QBLIT aufrufen
    move.l   (a7)+,a6     ;A6 zurückholen
    bsr      waitmessage  ;Auf Reply-Message warten
    movem.l  -22(a2),a1/a0/d0 ;Register zurückholen
    move.l   d0,d1        ;Anzahl der Bytes nach D1
    move.l   a1,a0        ;Zeiger auf Anfang der Daten
                          setzen
    bsr      connect      ;Rand korrigieren
    adda.l   d1,a0        ;Zeiger auf nächste Nahtstelle
    bsr      connect      ;Rand korrigieren
    adda.l   d1,a0        ;Zeiger auf Ende
    bsr      connect      ;Rand korrigieren
    unlk     a2           ;Platz der Struktur wieder
                          freigeben
    rts                   ;Rücksprung
qbroutine:
;Routine, die von der QBLIT-Funktion angespr.wird
;A0 zeigt auf $DFF000
;A1 zeigt auf die zuvor erstellte Struktur
;A6 zeigt auf die Trackdisk-Device-Struktur
    move.l   a5,-(a7)     ;A5 retten
    move.l   a1,a5        ;Zeiger auf Struktur kopieren
    bsr      setmodulo    ;Blitterparameter setzen
    move.l   a5,a1        ;A1 zurückholen
    movem.l  8(a1),a5/d1/d0 ;Zeiger auf Quelle und
                             Ziel holen
    move.l   d1,76(a0);Quelle nach BLTBPT (Source A)
    move.l   d1,80(a0) ;sowie nach BLTAPT (Source B)
    move.l   a5,84(a0) ;Ziel nach BLTDPT
                        (Destination D)
    move.w   #$1db1,64(a0)  ;Wert für BLTCON0
                             (Area Modus)
    ;DMA-Bits für A,B,D
    ;Verschiebung um 1 nach rechts für A
    ;ABC+AbC+Abc+abc als Minterm
    move.w   #0,66(a0);Wert für BLTCON1 (Area Modus)
    move.w   20(a1),88(a0) ;BLTSIZE beschreiben
                            (Blitterstart)
    move.l   #qbroutine2,4(a1) ;Zeiger auf nächste
                                Funktion, die  →
```

```
;nach dieser ausgeführt werden soll
    move.l  (a7)+,a5        ;A5 zurückholen
    rts                     ;Rücksprung

qbroutine2:
;Routine wird aufgerufen, wenn Blit-Vorgang der ersten Funktion
;beendet ist. Parameter sind, wie bei folgenden Routinen, gleich.
    move.l  a5,-(a7)        ;A5 retten
    movem.l 8(a1),a5/d1/d0  ;Werte aus der Struktur holen
    move.l  a5,76(a0)       ;Source B = Ziel
    move.l  d1,80(a0)       ;Source A = Quelle
    move.l  a5,84(a0)       ;Destination D = Ziel
    move.w  #$2d8c,64(a0)   ;Wert für BLTCON0
                            ;DMA-Bits für A,B,D
;Verschiebung um 2 nach rechts für A
;ABC+aBC+aBc als Minterm
    move.w  20(a1),88(a0)   ;BLTSIZE beschreiben (Blitterstart)
    move.l  #qbroutine3,4(a1) ;Zeiger auf nächste Funktion
    move.l  (a7)+,a5        ;A5 zurückholen
    rts                     ;Rücksprung

qbroutine3:
    move.l  a5,-(a7)        ;A5 retten
    movem.l 8(a1),a5/d1/d0  ;Werte aus der Struktur holen
    add.l   d0,d1         ;Zeiger auf Ende der Quelle
    subq.l  #2,d1         ;Ende -2
    adda.l  d0,a5         ;Zeiger auf Ende
    adda.l  d0,a5         ;des Ziels setzen
    subq.l  #2,a5         ;Ende -2
    move.l  d1,76(a0)     ;Source B zeigt auf Ende der Quelle
    move.l  d1,80(a0)     ;Source A zeigt auf Ende der Quelle
    move.l  a5,84(a0)     ;Destination D zeigt auf Titelende
    move.w  #$0db1,64(a0) ;Wert für BLTCON0
;DMA-Bits für A,B,D
;keine Verschiebung
;ABC+AbC+Abc+abc als Minterm
    move.w  #$1002,66(a0)   ;Wert für BLTCON1
;Verschiebung um 1 für B
;Descent-Mode (Vom Ende zum Anfang)
    move.w  20(a1),88(a0)   ;BLTSIZE beschreiben (Blitterstart)
    move.l  #qbroutine4,4(a1) ;Zeiger auf nächste Funktion
    move.l  (a7)+,a5        ;A5 zurückholen
    rts                     ;Rücksprung

qbroutine4:
    move.l  a5,-(a7)        ;A5 retten
    movem.l 8(a1),a5/d1/d0  ;Werte aus der Struktur holen
    adda.l  d0,a5           ;Zeiger auf Ende des ersten Zeilteils
    move.l  a5,76(a0)       ;als Source B eintragen
    move.l  d1,80(a0)       ;Source A = Quelle
    move.l  a5,84(a0)       ;Destination D = Ende des ersten Teil vom Ziel
    move.w  #$1d8c,64(a0)   ;Wert für BLTCON0
;DMA-Bits für A,B,D
;Verschieb. um 1 nach rechts für A
;ABC+aBC+aBc als Minterm
    move.w  #0,66(a0)       ;Wert für BLTCON0
    move.w  20(a1),88(a0)   ;BLTSIZE beschreiben (Blitterstart)
    move.l  #qbroutine5,4(a1) ;Zeiger auf nächste Funktion
    move.l  (a7)+,a5        ;A5 zurückholen
    rts                     ;Rücksprung

qbroutine5:
    moveq   #0,d0           ;D0 löschen, keine weitere Funktion)
    move.l  d0,4(a1)        ;Funktionszeiger löschen
    move.l  26(a1),a1       ;Zeiger auf Laufwerks-Port holen
    bsr     sendmessage     ;Message für Blit-Ende senden
    moveq   #0,d0           ;Flag für Aussprung aus QBLIT-Funkt.
    rts                     ;Rücksprung

setmodulo:
;Routine setzt die Modulo-Register u.Masken für
;das erste und letzte Wort
    moveq   #0,d0           ;D0 löschen
    lea     68(a0),a1       ;Zeiger auf BLTAFWM
    move.l  #-1,(a1)        ;BLTAFWM und BLTALWM setzen
    lea     98(a0),a1       ;Zeiger auf BLTBMOD
    move.l  d0,(a1)+        ;BLTBMOD und BLTAMOD löschen
    move.w  d0,(a1)+        ;BLTDMOD löschen
    addq.l  #8,a1           ;Zeiger auf BLTCDAT setzen
    move.w  #$5555,(a1)     ;BLTCDAT setzen
    rts                     ;Rücksprung

sendmessage:
;Routine sendet Rückmeldung zum Beenden des Wartevorgangs
;A1 zeigt auf den Laufwerks-Port
;A6 zeigt auf die Trackdisk-Device-Struktur
    lea     174(a1),a0      ;Zeiger auf Reply-Port
    lea     94(a1),a1       ;Zeiger auf Reply-Message
    move.l  a6,-(a7)        ;A6 retten
    move.l  execbase,a6     ;Zeiger auf Execbase
    jsr     putmsg(a6)      ;Message senden
    move.l  (a7)+,a6        ;A6 zurückholen
    rts                     ;Rücksprung

waitmessage:
;Routine wartet, bis eine Message zum Reply-Port gesandt wird
;A3 zeigt auf den Laufwerks-Port
;A6 zeigt auf die Trackdisk-Device-Struktur
    move.l  #$400,d0      ;SignalSet als Parameter der WAIT-Funktion
    move.l  a6,-(a7)        ;A6 retten
    move.l  execbase,a6     ;Zeiger auf Exexbase
    jsr     wait(a6)        ;Task in Warte-Status setzen
    move.l  (a7)+,a6        ;A6 zurückholen
    lea     174(a3),a0      ;Zeiger auf Reply-Port
    move.l  a6,-(a7)        ;A6 retten
    move.l  execbase,a6     ;Zeiger auf Execbase
    jsr     getmsg(a6)      ;Message holen
    move.l  (a7)+,a6        ;A6 zurückholen
    tst.l   d0              ;Message angekommen?
    beq     waitmessage     ;Nein, weiter warten
    rts                     ;Rücksprung

psumme:
;Berechnet die Prüfsumme (Byteanzahl muß durch 4 teilbar sein!!!)
;A0 zeigt auf die Daten, deren Prüfsumme berechnet werden soll
;D1 enthält die Anzahl der Daten in Bytes
    move.l  d2,-(a7)        ;D2 retten
    lsr.w   #2,d1           ;Byteanzahl durch 4 teilen
    subq.w  #1,d1           ;wegen dbf
    moveq   #0,d0           ;Ergebnis auf Null setzen
pmarke:
    move.l  (a0)+,d2          ;Langwort holen
    eor.l   d2,d0             ;und verknüpfen
    dbf     d1,pmarke         ;bis zum Ende
    and.l   #$55555555,d0   ;ungültige Bits entfernen
    move.l  (a7)+,d2          ;D2 zurückholen
    rts                       ;Rücksprung

trdname: dc.b "trackdisk.device",0
even
quelle: blk.b 512,0
ziel: blk.b 1090,0
```

*Listing 8*

→

```
;Listing 9
decodeblock:
;A0 zeigt auf Bereich, in den decodierten Daten
  geschr.werden
;A1 zeigt auf Bereich, in dem die zu decodieren-
  den Daten stehen
;A3 zeigt auf den Laufwerks-Port
;A6 zeigt auf die Trackdisk-Device-Struktur
;D0 enthält die Anzahl zu decodierender Bytes

 link    a2,#-30        ;Platz für QBLIT-Struktur
 movem.l a1/a0/d0,-22(a2) ;Werte in Struktur
                          eintragen
 lsl.w   #2,d0          ;Wert für
 or.l    #8,d0          ;BLTSIZE berechnen
 move.w  d0,-10(a2)     ;und in Struktur eintragen
 move.l  #decoroutine1,-26(a2) ;Zeiger auf
                               Funk.eintragen
 move.l  a3,-4(a2)      ;Laufwerks-Port eintragen
 lea     -30(a2),a1     ;Zeiger auf Strukturanfang
 move.l  a6,-(a7)       ;A6 retten
 move.l  56(a6),a6      ;Zeiger auf GfxBase
 jsr     qblit(a6)      ;QBLIT-Funktion aufrufen
 move.l  (a7)+,a6       ;A6 zurückholen
 bsr     waitmessage    ;Auf Reply-Message warten
 movem.l -22(a2),a1/a0/d0 ;Register zurückholen
 unlk    a2    ;Speicher der Struktur freigeben
 rts                    ;Rücksprung

decoroutine1:
;Funktion wird von der QBLIT-Funk.aufgerufen,
 wenn der
;Blitter frei ist.
;A0 zeigt auf $DFF000
;A1 zeigt auf die QBLIT-Struktur
;A6 zeigt auf die Trackdisk-Device-Struktur
 move.l  a5,-(a7)    ;A5 retten
 move.l  a1,a5 ;Zeiger auf QBLIT-Struktur merken
 bsr     setmodulo ;Blitter Parameter setzen
 move.l  a5,a1       ;Zeiger auf Struktur holen
 movem.l 8(a1),a5/d1/d0  ;Zeiger auf Quelle, Ziel
                          und Länge
 adda.l  d0,a5       ;Quellenende für ungerade Bits
 subq.l  #1,a5       ;weniger eins
 move.l  a5,80(a0)   ;BLTAPT
 adda.l  d0,a5       ;Endadresse der geraden Bits
 move.l  a5,76(a0)   ;BLTBPT
 add.l   d0,d1       ;Endadresse + Länge
 subq.l  #1,d1       ;weniger eins
 move.l  d1,84(a0)   ;BLTDPT
 move.w  #$1dd8,64(a0)  ;Wert für BLTCON0
 ;Verschiebung um 1 nach rechts für A
 ;DMA-Bits für A,B und D
 ;ABC+ABc+Abc+aBC als Minterm
 move.w  #2,66(a0)   ;Wert für BLTCON1
                     ;Descent-Mode
 move.w  20(a1),88(a0) ;BLTSIZE beschreiben
                        (Blitterstart)
 move.l  #decoroutine2,4(a1) ;nächste Funk.setzen
 move.l  (a7)+,a5            ;A5 zurückholen
 rts                         ;Rücksprung

decoroutine2:
 moveq   #0,d0        ;D0 löschen
 move.l  d0,4(a1)     ;Zeiger auf Funk.löschen
 move.l  26(a1),a1    ;Zeiger auf Laufwerks-Port
 bsr     sendmessage  ;Message für Ende senden
 moveq   #0,d0        ;Keine weitere Funk.aufrufen
 rts                  ;Rücksprung
```

*Listing 9*

VON INGO BRÜMMER

# Der KEIL




## *Toshiba P341SL -24 Nadeln für den Profi*

*Schick schaut er aus, der Toshiba. Keine graue Alltagskiste bevölkert den Schreibtisch, sondern ein schnittiges Gehäuse schickt sich an, mein Papier zu bedrukken. Seine Ausstattung, seine Verarbeitung und sein Preis deuten darauf hin, daß Toshiba mit ihm potentielle Käufer im Bürobereich sucht.*

Toshiba ist ein japanischer Multi-Konzern. Es gibt praktisch nichts Elektronisches, das es nicht auch von dieser Firma gäbe. Doch daß aus dem fernöstlichen Haus auch Drucker kommen, das ist nicht sonderlich bekannt. Der P341SL ist schon eine Zeit lang am Markt und steht uns für die folgenden Seiten zu einem Test zur Verfügung.

### Saubere Endlosverarbeitung

Zunächst aber muß das Gerät ausgepackt und aufgebaut werden. Die nötigen Montagearbeiten sind schnell getan. Der Anschluß an den Computer findet über die parallele Schnittstelle statt; ohne Aufpreis steht aber auch eine serielle (RS 232-C) zur Verfügung.

Die Oberseite des P341SL wird abgedeckt durch zwei Klappen. Die vordere der beiden ist zweigeteilt und läßt sich in zwei Stufen öffnen. Zunächst gibt der Rauchglasteil den Blick auf Walze und Kopf frei, dann öffnet der Rest den gesamten Druckbereich. (Wie sich später zeigen soll, druckt das Gerät bei offener Haube weiter.) Die hintere Klappe bedeckt den Schubtraktor und braucht nur beim Wechseln des Endlospapiers geöffnet zu werden. Auf ihr wiederum befindet sich - hochstellbar - die Rutsche für die Einzelblätter. Die beschriebene Bauart von Gehäusen setzt sich immer mehr durch, da so die Hauben fest am Gerät angeschlagen werden können. Dementsprechend fällt auch beim To-

shiba kein Teil beim Öffnen ab, und man hat nie irgendwelches Zubehör auf dem Schreibtisch liegen.

Der erste Eindruck von der Verarbeitung eines Druckers entsteht fast immer beim Einlegen des Endlospapiers. So auch beim Toshiba P341SL. Die Stacheltraktoren, die unter der hinteren Klappe zum Vorschein kommen, sind längere Bänder und keine Räder mehr wie bei anderen - vor allem älteren - Druckern. Das hat den Vorteil, daß das Papier weitaus besser geführt wird. Gerade für die Papierparkfunktion, über die auch der P341SL verfügt, ist das wichtig. Denn beim Wechsel von Endlospapier auf Einzelblatt muß das Papier vom Drucker zurückgefahren werden. Passieren dabei Fehler, gibt es ein heilloses Chaos, denn die Drucker quetschen das Papier dann in der Mechanik zusammen. Genau das aber ist im Verlauf des gesamten Tests nicht ein einziges Mal geschehen, das Papier wurde von den Traktoren immer sicher vor- und zurückgefahren.

Wie bereits erwähnt, ist der P341SL in der Lage, das Endlospapier zu parken, die Prozedur des Papiereinlegens braucht also nur höchst selten vorgenommen zu werden. Es sei denn, man will nun Aufkleber oder anderes perforiertes Druckgut bearbeiten.

Der Wechsel zwischen den Papiersorten ist sehr einfach: Befindet sich Endlospapier im Drucker, braucht nur der Hebel auf der rechten Seite des Gerätes nach vorn gezogen zu werden. Geschieht das im Off Line-Status, wird das Papier zurückgefahren. Daraufhin schaltet man mittels des Hebels links auf Einzelblattverarbeitung und kann das erste Blatt einziehen. Das geschieht wiederum mit dem rechten Hebel: Papier auf die Rutsche legen, Hebel ziehen, fertig. Insgesamt zeigt der Toshiba ein problemloses Papierhandling. Einzig die seitlichen Führungen für das Einzelblatt, die sich auf der Rutsche befinden, sind ein wenig dürftig. So manches Blatt muß man gleich noch einmal einziehen, weil es schief hinter der Walze gelandet war.

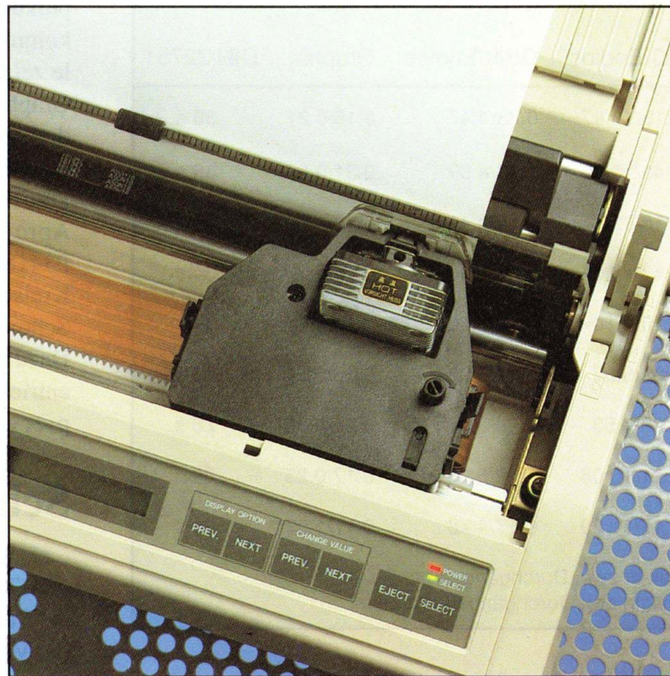


*Bild 1: Das Bedienerfeld kontrolliert sämtliche Einstellungen*



*Bild 2: Dr. Toshiba und Mr. Hide-die Emulationskarten*

Auffällig am Toshiba P341SL ist der Umstand, daß nur sechs Folientasten sein Bedienfeld zieren. Zwei LEDs dazu und ein LCD-Display - von anderen Herstellern kriegt man zuweilen ein Cockpit geboten. Das wäre aber weder dem schlichten Design noch der Übersichtlichkeit zuträglich gewesen. Daher werden alle Einstellungen des Druckers - von der Wahl der Schnittstelle bis zum Zeilenabstand - mit den Tasten vorgenommen. Die LCD-Anzeige informiert währenddessen. Die Menüs, die der Drukker dabei zutage fördert, sind äußerst umfangreich und glücklicherweise im Handbuch graphisch verdeutlicht.

Dabei gibt es einige, für den Bürobetrieb recht nützliche, Features. Zum einen kann aus sämtlichen Menüpunkten ein kleines Menü zusammengestellt werden. So erlaubt man dem Benutzer nur die Veränderung weniger Parameter, wie zum Beispiel der Druckqualität. Man könnte es auch anders ausdrücken: Für ihn unwichtige Dinge werden dem Bedienenden vom Hals gehalten. In das Hauptmenü ist denn auch nur mit einem Trick zu gelangen. Eine andere Möglichkeit besteht darin, komplette Zusammenstellungen von Parametern anzuwählen. Leider gibt es davon nur vier vorgegebene, die sich nicht ändern lassen. Es ist also kein Makrobetrieb vorgesehen, bei dem man derartige Gesamteinstellungen selbst verändern und mit wenigen Handgriffen auswählen kann. Selbstverständlich gibt es ein Menü, in dem man alle Einstellungen so verändert, daß sie beim Einschalten des Druckers wirksam sind. Die Zeiten der früher üblichen DIP-Schalter sind also vorbei.

Das Handbuch ist übrigens sehr übersichtlich und anschaulich gestaltet (Spiralbindung!). Es hat nur einen Nachteil: Es ist in englischer Sprache verfaßt. Schade, daß es noch Firmen gibt, die auf des Deutschen sprachliche Anpassungsfähigkeit spekulieren und selbst bei Geräten für über 3000 DM ein Handbuch nicht übersetzen.

## Zeitvergleich

*Dokumentlänge Text: 33396 Bytes*
*Dokumentlänge Graphik: 32643 Bytes*
*Genannt: Zeit zur Datenabnahme / Gesamtzeit für Druck*

| Drucker | LQ-Einzel | LQ-Endlos | Draft-Endlos | Graphik | DIN 32751 |
|---|---|---|---|---|---|
| Brother M-1724L | 0:18/12:02 | 0:18/10:47 | 0:23/4:47 | 0:16/0:21 | 38 s |
| Citizen HQP-45 | - | 2:48/10:39 | 1:16/4:59 | 0:31/0:32 | 38 s |
| C.Itoh C-715A | - | 0:18/8:31 | 0:19/3:41 | 0:18/1:02* | 32 s |
| Epson LQ-2500 | - | 7:47/10:04 | 3:49/4:56 | 0:16/0:17 | 34 s |
| NEC P6 PLUS | - | 0:18/9:10 | 0:18/5:46 | 0:16/0:20 | 34 s |
| OKI ML 393C | 0:50/7:46 | 0:39/6:27 | 0:20/3:02 | 0:16/0:28 | 26 s |
| OKI ML 390 | - | 5:20/7:53 | 2:20/3:27 | 0:24/0:25 | 28 s |
| Seikosha SL-130AI | - | 5:21/10:08 | 2:42/5:06 | 0:17/0:25 | 36 s |
| Toshiba P341 SL | - | 0:17/11:29 | 0:17/4:48 | 0:17/1:03** | 41 s |

* Der C-715A druckt 180 DPI-Graphik in zwei Durchgängen pro Zeile.
** Der P341SL druckt 180 DPI-Graphik in drei Durchgängen pro Zeile.

# Die Software

Der Toshiba P341SL bietet bei seiner Auslieferung zwei Emulationsarten: Toshiba/Qume und IBM. Zwischen diesen beiden ist per Software-Befehl oder Voreinstellung zu wählen. Doch hat die Sache einen Haken: Wählt man IBM-Emulation, hat man mit entsprechenden Treibern in Sachen Textverarbeitung keine Probleme. Jedoch sieht es schlecht mit der Graphik aus, denn der emulierte IBM-Drucker ist ein 9-Nadler. Man muß sich dann also mit Acht-Nadel-Graphik begnügen. Wählt man Toshiba-Emulation, hat man wieder Zugang zu echter 24-Nadel-Graphik. Leider gibt es jedoch wenige Treiber für diese Kommandos. Der Toshiba-Befehlssatz hat nämlich partout keine Ähnlichkeit mit dem von IBM oder Epson.

Findet man nun aber keinen Treiber für den Toshiba-Modus, kommt einer der beiden Schlitze auf der Vorderseite des P341SL zur Geltung. Verfügbar ist nämlich eine Karte, die man dort einschieben kann und die es dem Gerät ermöglicht, einen Epson LQ-Drucker zu emulieren. Nachdem diese Karte eintraf, gab es auch keine Schwierigkeiten auf der Softwareseite mehr.

Leider bleibt noch ein Punkt, der sich

```
Hoch auf dem gelben Wagen
Hoch auf dem gelben Wagen
Hoch auf dem gelben Wagen
```

*Bild 3: Schriftprobe des Toshiba P341SL*

360 x 180 DPI

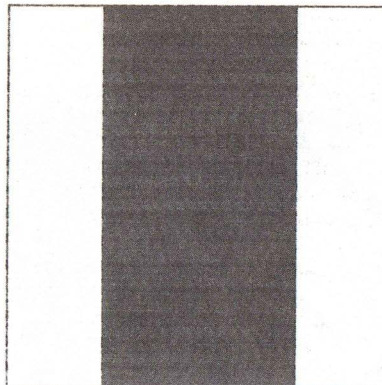


*Bild 4: Grafikprobe (unidirektional gedruckt) des Toshiba P341SL*

auch bei den Zeitmessungen widerspiegelt. Es war dem P341SL nicht abzugewöhnen, Graphiken in drei Durchgängen zu drucken. Zwar kann man das im Menü abstellen, doch beschränkt sich die Wirksamkeit dieser Einstellung offensichtlich auf die festen Emulationen. Sobald der Drucker im Epson-Modus betrieben wurde, druckte er wieder in drei Durchgängen und grundsätzlich unidirektional. Das verzögert den Graphikdruck natürlich ungemein, und so kommt die schlechte Zeit, die die Tabelle zeigt, zustande. Gerade fortgesetzter Graphikdruck wird zur Qual: Seitenlang drei Durchgänge pro Zeile und das mit einem recht lauten Gerät, das ist was für Frauen und Männer ohne Nerven.

Apropos Geräusch: Der Toshiba P341SL gibt ein subjektiv störendes Kreischen von sich, das genau die Schmerzgrenze meines Ohres trifft. Dazu kommt der recht laute Druckkopfantrieb und der ebensowenig leise Papiervorschub.

# Schneller...

...als mit anderen Druckern ist man mit dem Toshiba nicht, wie Sie sicher schon der Tabelle entnommen haben. Sie zeigt die Vergleichszeiten von Druckern, mit denen sich der P341SL messen lassen muß. Das betrifft vor allem das Preisniveau, viele der Konkurrenten kosten sogar weniger. Alle Zeiten wurden mit einem Druckertestprogramm ermittelt. Die Spalte 'DIN 32751' kann als Anhaltspunkt für die Druckdauer einer DIN A4-Seite dienen, denn unter diesem Testpunkt wird ein Geschäftsbrief gedruckt. Auffallend am Toshiba jedoch ist die hohe Geschwindigkeit, in der er dem Rechner die Daten abnimmt. Das liegt an seinem Speicher von 32 kByte. Einen Spooler einzusetzen, wird dabei kaum nötig sein.

# Besser...

...als bei anderen 24-Nadlern ist die Draft-Qualität des P341SL. Aufgrund des geringen Nadeldurchmessers ist diese immer schlechter als bei 9-Nadlern, denn um schnell zu sein, werden auch nur acht Nadeln benutzt. Der Toshiba fällt da angenehm auf, wie Sie in der entsprechenden Abbildung sehen können. Auch die Graphikqualität ist gut, die senkrechten Linien sind sehr exakt, was auf eine sauber verarbeitete Mechanik zurückzuführen ist. Die abgebildete Testgraphik ist mit unserem Druckertestprogramm erstellt worden. Sie können daran die Qualität senkrechter Linien, schwarzer Flächen und die

Maßhaltigkeit von Graphiken (Länge und Breite) erkennen. Die maximale Auflösung des P341SL beträgt 360 x 180 Punkte pro Zoll waagerecht mal senkrecht.

Die Briefqualität ist normal, zur Verfügung steht normalerweise nur der eingebaute Font Courier. Wer's vielfältiger möchte, kann weitere Fonts auf Steckkarten hinzukaufen. Damit ist jetzt auch das Geheimnis des zweiten Schlitzes vorn im Drucker gelüftet. Eine Emulations-und eine Fontkarte oder zwei Fontkarten, das ist die maximale Bestückung unseres Kandidaten.

Ob der Toshiba P341SL insgesamt die bessere Wahl ist, das muß natürlich wieder einmal der Verbraucher selbst entscheiden. Fest steht, daß dies ein Drucker ist, der für professionellen Einsatz am IBM PC gebaut wurde. Die Menüführung und der Umstand, daß er serienmäßig keine Epson-Emulation bietet, weisen darauf hin. Seine Geschwindigkeit ist allerdings insgesamt mäßig, die Qualität der Verarbeitung deutet wiederum darauf hin, daß er für mehr als nur einen Brief pro Tag geschaffen ist. Die Preise, die Toshiba für das Gerät verlangt, befinden sich ebenfalls auf einem Niveau, das deutlich in Richtung Geschäft zeigt. Leider ist noch nicht einmal bei einem 3000 DM-Drukker der Kopf Bestandteil der Garantie und das Handbuch zudem noch englisch.

## TOSHIBA P341SL

+ gute Verarbeitung
+ sichere Papierverarbeitung
+ übersichtliche Bedienung

- englisches Handbuch
- Epson-Emulation nur gegen Aufpreis
- hoher Preis
- laut

| | |
|---|---|
| *Preis:* | *3066 DM* |
| *autom. Einzelblatteinzug:* | *900 DM* |
| *Farbband:* | *25 DM* |
| *Emulationskarte:* | *235 DM* |
| *Fontkarte:* | *165 DM* |

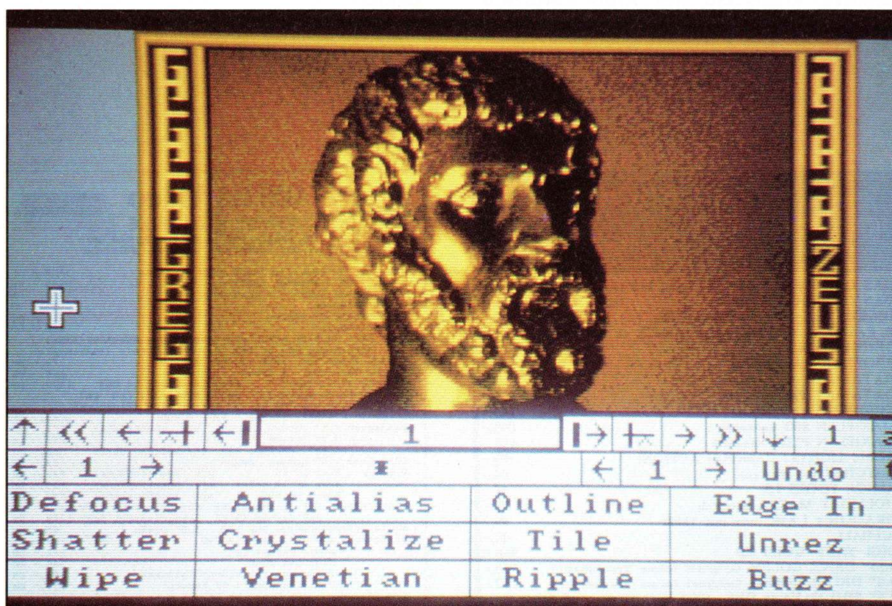
VON WILFRIED SOLBACH

# Lebensräder für alle

# ZOETROPE

## 3D-Animation-Echtheit

*Zoetrope (sprich: zoui,troup) ist der englische terminus technicus für einen "stroboskopischen Zylinder", ein Gerät aus den Kindertagen der Bildanimation, dessen Geschichte weit in das 19. Jahrhundert zurückgeht.*




*Vielfälltige Bildmanipulationen sind mit ZOETROPE kein Problem.*

Ein Artikel aus dem Jahre 1910(!) faßt ziemlich genau die Grundidee des Animationsprogramms aus dem kalifornischen Software-Haus Antic zusammen:

"...the zoetrope or 'wheel of life', -described by W.G. Horner around 1833, which consists of a hollow cylinder turning on a vertical axis and having its surface pierced with a number of slots. Round the interior is arranged a series of pictures representing successive stages of such a subject as a galloping horse..."

Auf Verpackung und Handbuch von Zoetrope ist eben dieses abgebildet: ein sich entrollender Filmstreifen, dessen Einzelbilder die Bewegungsphasen eines galoppierenden Pferdes darstellen. Die Art der Darstellung ist eine klare Anlehnung an die Originalbilder aus dem 19. Jahrhundert.

Aber keine Angst, das Programm kann einiges mehr. Es nennt sich im Untertitel "The Animation System" - es handelt sich nach Aussagen des Herstellers um "three animation programs in one", dazu unten Näheres.

## Was gibt es fürs Geld?

Der Lieferumfang umfaßt zwei Disketten. Auf der eigentlichen Programmdiskette befinden sich neben dem Hauptprogramm ein frei kopierbarer "Player" und die Konvertierungsprogramme ANITOZOE und CROPPER. ANITOZOE konvertiert 'Aegis Animator-.script'-Dateien in das 'Zoetrope .rif-Format', CROPPER verwandelt u.a. ANIM-Dateien (Videoscape 3D, Video Titler) und 'Atari Cyber Paint-files' in das .rif-Format. Außerdem können Zoetrope-files in ANIM und .script umgewandelt werden. Auf der Datendisk befinden sich des weiteren einige Animationen und Hintergrundbilder.

Das etwa 180 Seiten starke Handbuch ist mit das Beste, was ich bisher für den AMIGA gesehen habe: Ein kurzer Einführungsteil für die, die sowieso schon alles wissen (Quick Start), ein sechsteiliges Tutorial, das in die einzelnen Funktionen des Programms Schritt für Schritt einführt, und ein Referenzteil, der die einzelnen Menüpunkte wie ein Lexikon erläutert und ergänzende Informationen zum Lernteil liefert. Die Düs-

seldorfer Firma GFA plant für Anfang Mai die Herausgabe der PAL-Version mit deutschem Handbuch.

Das englische Handbuch, das mir zur Verfügung stand, ist einfach ohne Schnörkel und Fachidiotien, geschrieben, manchmal locker und humorvoll. Es enthält alle wichtigen Informationen, um in kurzer Zeit eindrucksvolle Animationen zustande zu bringen. Das Programm als solches ist nicht kopiergeschützt, ist aber ohne das Handbuch kaum zu gebrauchen, da es sehr komplex in der Befehlsabfolge ist und darüberhinaus noch über sogenannte "hidden commands" verfügt, auf die man auch bei jahrelanger Probiererei nicht kommen kann. Wie heißt es so schön: Der effektivste Kopierschutz ist ein gutes Handbuch.

Startet man das Hauptprogramm, so empfängt einen eine gähnende schwarze Leere, in die ein Mausklick weiße Pulldownmenüs zaubert: die optische Ähnlichkeit mit Aegis Animator läßt sich nicht verleugnen, schließlich hat Jim Kent beide Programme geschrieben.

Das Programm verfügt über LowRes-Auflösung, andere Wahlmöglichkeiten sind nicht gegeben. Jeder Befehl läßt sich durch Drücken der rechten Maustaste abbrechen, was die Arbeit sehr komfortabel macht. Auch die Tastaturbelegung ist sehr intelligent, da die meisten Befehle durch Drücken der Anfangsbuchstaben aufgerufen werden können (also z.B. D = Draw).

## Stärken

Auf dem unteren Bildschirmrand erscheint eine Leiste, die Ähnlichkeit mit den Tasten eines Videorekorders hat. Es gibt Knöpfe für Vorlauf, Rücklauf, erstes/letztes Bild etc. Beim Programmstart verfügt man nur über ein leeres Bild, das in der späteren Animation einem "frame" oder Standbild entspricht. Durch <Return> läßt sich die Anzahl der frames erhöhen.

Ein komfortables Zeichenprogramm macht es leicht, die optischen Bausteine für die Animation zu erstellen. Die meisten der aus anderen Zeichenprogrammen bekannten Features finden sich auch hier, es läßt sich aber zum Beispiel die Farbpalette nicht nur im RGB-Modus verändern, sondern zusätzlich noch im HLS-Bereich (hue, luminance, saturation), was interessante Pastelltöne ermöglicht. Die Geschwindigkeit der "Spraydose" läßt sich stufenlos verstellen, außerdem kann auch mit Brushes oder selbstgewählten Clips gesprüht werden. Bilder und Bildteile können problemlos ausgeschnitten und auf den gesamten Film mit Hilfe von "paste" übertragen werden. Das Zeichenprogramm enthält noch ein paar Überraschungen, z.B. beim Erstellen von Kreisen oder Polygonen, die aber hier nicht verraten werden sollen.

IFF-Graphiken lassen sich einladen und bearbeiten: Dazu steht nicht nur das Malprogramm zur Verfügung, sondern auch eine butcherähnliche Nachbearbeitungsfunktion, die aber einige neue Features aufweist: OUTLINE, SHATTER, CRYSTALIZE und EDGE IN sorgen für atemberaubende Effekte. Schon hier zeigt sich die Geschwindigkeit des Programms, das Graphiknachbearbeitung bedeutend schneller als z.B. Butcher erledigt.

Der eigentliche Clou des Programms ist der APM oder Antic Pixel Mover. Hier lassen sich vorher im "clip buffer" abgelegte Bild-/Textpassagen um bis zu drei




*Der 'AEGIS ANIMATOR CLOWN' kann in das ZOETROPE-Format umgewandelt werden.*




*Werbedemos und ähnliches lassen sich leicht mit dem Programm erstellen.*

Achsen drehen, auf dem Bildschirm bewegen, Einzelbilder lassen sich beliebig in eine Sequenz einfügen und löschen, erstellte Animationen können durch eine "preview"-Funktion begutachtet und, indem man den "Weg" mit der Maus auf den Bildschirm zeichnet, in ihrer Richtung geändert werden. Die Befehlsvielfalt und ihre Kombinationen erschlagen den Benutzer im ersten Augenblick, hat man sich eingearbeitet, läßt APM (das MAZ-Misch- und Trickpult des Programms) auch den Animationsanfänger professionell wirkende Animationen erstellen. Aber auch hier gilt: ohne Handbuch, nothin' doin'.

Für die fließende Animation sorgt vor allem das rif.-Format, das die einzelnen Animationsteile komprimiert ablegt und nur die Unterschiede zwischen den einzelnen "Frames" registriert. Die eigentliche Animation findet dann in Echtzeit statt.

Das Programm benötigt mindestens 1MB Speicher, aber der warnende Hinweis "Out of chip memory" kommt eher früher als später. Zwar lassen sich auch mit "nur" einem Megabyte RAM schon schöne Ergebnisse erzielen, aber zum frustlosen Arbeiten sind mindestens 3MB und eine Festplatte angeraten. Externe Laufwerke sollten abgeschaltet werden, man benötigt wirklich jedes Byte. Das Handbuch enthält noch eine Reihe weiterer Spartips.
Um kostbaren Speicher zu sparen, empfiehlt es sich, längere Animationen in einzelnen Teilen zu produzieren und abzuspeichern. Über das CLI läßt sich QUICKRIF aufrufen, eine Art slideshow, die verschiedene Animationsfiles hintereinander abspielt. Ein anderer "Speicherspartrick" besteht darin, in der Nachbearbeitungsphase nur Teile bestimmter Animationsdateien ("Frame 34-41") einzuladen und diese nach erfolgreichem "fine editing" wieder im Ursprungsfile an ihrem alten Platz abzuspeichern.

Während des Tests wurde die Aegis-Animation "Clown", die etwa 85 KB groß ist, in das .rif-Format konvertiert, das Ergebnis: 320 KB, 504 Frames und eine fließendere Animation.

## Fazit

Bei Zoetrope 1.1 handelt es sich um ein sehr leistungsstarkes und bedienerfreundliches Animationsprogramm, das an Schnelligkeit der Berechnung und fließender Animation seinesgleichen sucht. Viele andere interessante Eigenschaften und Stärken von Zoetrope - insbesondere im Bereich Frame-Bearbeitung und Schnitt konnten aus Platzgründen nicht einmal erwähnt werden. Der Preis von DM 198.- für die deutsche Version ist nicht zu hoch, bedenkt man die unübersehbaren Stärken der Software. Erfreulich ist zudem, daß das Programm erheblich preiswerter als in den USA angeboten wird. Außerdem ist von seiten des Programmierers geplant, in künftigen Updates Musik und Geräusche in die Animationen miteinbinden zu können.

**_ZOETROPE 1.1_**
**_The Animation System_**

+ schnelle, fließende Animation
+ durchdachte Befehle, short cuts
+ lädt und konvertiert andere Formate (z.B. Aegis und Atari)
+ gut gemachtes, deutsches Handbuch
+ Animation in Echtzeit
+ PAL

- nur eine Auflösung (320x256)
- keine Sound-Einbindung
- hoher Speicherbedarf

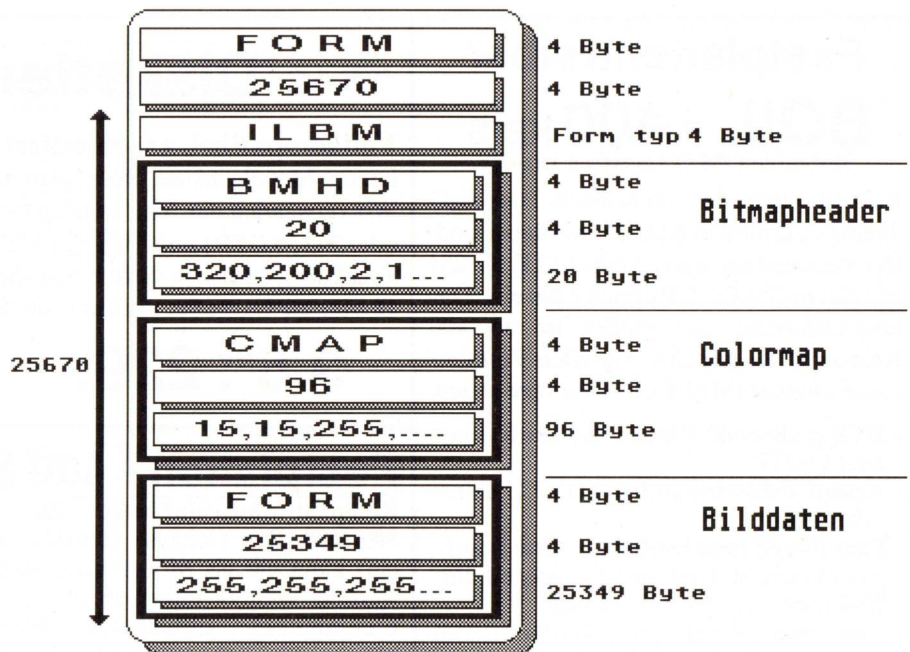


VON JÜRGEN HAAGE

# IFF im Griff

## Das Grafikstandardformat

*IFF steht für Interchange File-Format und die Bedeutung ist jedem AMIGA-Anwender bekannt. Doch weder zum Laden noch zum Speichern von Grafiken stellt der AMIGA Funktionen zur Verfügung. Gerade Anfängern wird es dadurch schwer gemacht, IFF-Grafiken in eigenen Programmen zu nutzen, da das Format in den AMIGA-Handbüchern nicht beschrieben wird.*




Der IFF-Standard wurde geschaffen, um ein und dieselbe Datei in verschiedenen Programmen oder auf verschiedenen Rechnern ohne Konvertierung nutzen zu können. Speziell bei Grafikprogrammen hat sich der IFF-Standard durchgesetzt. Durch die Vielzahl der Grafikauflösungen, die der AMIGA bietet, ist man gezwungen, außer den eigentlichen Bildinformationen noch Weite, Höhe, Tiefe und eine Farbtabelle der Grafik mit abzuspeichern.

Eine IFF-Datei ist in einzelne Blöcke, sogenannte *"chunks"*, untergliedert. Jeder Block wird durch eine vier Byte große ID gekennzeichnet. Weitere vier Bytes stehen für die Länge des Blockes. Es folgen die Daten, entsprechend der angegebenen Länge. Durch diesen einfachen Aufbau ist es sogar möglich, eigene Informationen einzubauen, die von anderen Programmen nicht unbedingt genutzt werden müssen. In DeluxPaint werden so beispielsweise die Informationen für Farbeffekte in mehre-

ren Blöcken mit der ID *"CRNG"* abgespeichert, die zum reinen Anzeigen des Bildes einfach überlesen werden können.

In der Schemazeichnung sehen Sie den Aufbau einer IFF-Grafik. Durch die ersten 12 Bytes kann die Datei identifiziert werden. Trifft man, nachdem die ersten acht Bytes geladen sind, auf das Wörtchen *"ILBM"* (Interleaved Bitmap), handelt es sich eindeutig um eine IFF-Grafik. Der Block *"BMHD"* enthält die Größe und Tiefe der Grafik in folgendem Format (siehe nebenstehende Tabelle).

```
struct BitMapHeader {
  UWORD Weite          Weite der Bitmap         2 Bytes
  UWORD Höhe           Höhe der Bitmap          "
  WORD  X              X-Versatz der Grafik     "
  WORD  Y              Y-Versatz "  "           "
  UBYTE tiefe          Bitplanes                1 Byte
  UBYTE maske          Maskiertechnik           "
  UBYTE komprimiert    Komprimiertechnik        "
  UBYTE padbyte        ungenutzt                "
  UWORD transcolor     Transparent-Bitplanes    2 Bytes
  UBYTE XAspect        X-Ansicht                1 Byte
  UBYTE YAspect        Y-Ansicht                "
  WORD  Weite          Screenweite              2 Bytes
  WORD  Höhe           Screenhöhe               2 Bytes
};                                              _______
```

Die Weite und Höhe der Grafik ist unabhängig von der Screengröße und daher extra angegeben. Für Brushes kann so beispielsweise die Screengröße von 320 * 200 Pixel angegeben sein, die Bitmapweite und -höhe aber nur 64 * 80 Pixel betragen.

Die beiden Felder X und Y bestimmen die Position des Bildes innerhalb der Bitmap, was von vielen Programmen einfach ignoriert wird. Die Anzahl der Bitplanes für die Bitmap ist im Feld *tiefe* gespeichert.

Eine IFF-Datei mit einer Farbtabelle (Colormap), aber keinem *BODY* und einer Null in diesem Feld ist der beste Weg, eine Farbtabelle abzuspeichern.

Welche Maskiertechnik eingesetzt werden muß, wird im Feld *maske* festgelegt.

0 = keine Maske
1 = Eine Bitplane der Bilddaten enthält die Maske
2 = Pixel, die mit der in *transcolor* enthaltenen Farbnummer übereinstimmen, sollen transparent erscheinen.

Ein sehr großer Vorteil von IFF-Grafik liegt darin, die Bilddaten komprimiert speichern zu können, aber dennoch die Kompatibilität zu anderen Ladeprogrammen beizubehalten. Um die Grafik zum Anzeigen wieder dekomprimieren zu können, muß festgelegt werden, welcher Komprimieralgorithmus verwendet wurde. Ist im Feld *"komprimiert"* eine Null enthalten, sind die Bilddaten unkomprimiert. Eine Eins in diesem Feld bedeutet, daß die Bilddaten mit dem "byteRun1"-Komprimieralgorithmus komprimiert wurden, der im GFA-BASIC-Listing ab Zeile 140 zu finden ist.

Das Bitfeld *"transcolor"* legt fest, welche Bitplanes des Bildes transparent erscheinen sollen. Das Feld enthält dann einen Wert, wenn im Feld *"maske"* eine Zwei enthalten ist.

Das Seitenverhältnis des Bildes ist in *XAspect* und *YAspect* angegeben und beträgt für Low-Res Grafiken (320 * 200) 10:11.

Ein weiterer wichtiger Block ist die Farbtabelle, die durch die ID *"CMAP"* gekennzeichnet wird. Die Daten der einzelnen Farbregister werden in drei Bytes (ROT-GRÜN-BLAU) Werte abgelegt. Die Anzahl der zu füllenden Farbregister ergibt sich aus der "Länge des Blockes / 3". Die Intensität jeder Farbe reicht von 0 - 255 und muß demnach für den AMIGA umgerechnet werden, da er nur Intensitätswerte von 0 - 15 zuläßt. In der Procedur *"setcolors"* des GFA-BASIC-Listings wird verdeutlicht, wie die Farbwerte umgerechnet und in die Register geschrieben werden.

Die eigentlichen Bilddaten sind im Block *"BODY"* abgelegt. Die Zeilen jeder einzelnen Bitplane stehen hintereinander und müssen demnach folgendermaßen geladen werden.



*Das geladene Interlace-Bild benötigte mit dem BASIC-Lader ca. 3 Minuten. Der Fastloader hingegen nur 15 Sekunden.*

```
FOR zeile = 0 TO Höhe
    FOR planes = 0 TO tiefe-1
        BGET Bildzeile(planes),Weite
    NEXT planes
NEXT zeile
```

Ist das Bild komprimiert, kann es in Basic unter Umständen einige Minuten dauern, bis es komplett geladen ist. Schneller ginge es, wenn die einzelnen Bitplanes nacheinander immer als Ganzes geladen werden könnten, was das IFF-Format allerdings nicht zuläßt.

Mit der Procedur *"fastsave"* des Listings wird der angegebene Screen in dem schnelleren Format gespeichert. Mit der Procedur "fastload" kann dann eine so gespeicherte Datei sehr schnell wieder geladen werden. Aber bitte verwenden Sie dieses Format nur für Ihre eigenen Programme, da solche Bilddateien von keinem Zeichenprogramm geladen werden können.

K

```
' Iff-Lader und Schnelladeformat Umwandler
'
DIM bitplanes%(6)
CLR bmhd!,cmap!,body!
'
FILESELECT "IFF-ILBM laden","laden","",file$
IF LEN(file$)
  OPEN "I",#1,file$
ELSE
  END
ENDIF
grafbuf%=MALLOC(120,65537)
IF grafbuf%=0
  ALERT 0,"Nicht genug freier Speicher !",
        0,"OK",dummy&
  END
ENDIF
'
'
BGET #1,grafbuf%,12
IF MKL$({ADD(grafbuf%,8)})<>"ILBM"
  ALERT 0,"Keine IFF-ILBM Datei !",0,"OK",
        dummy&
  END
ENDIF
'
WHILE NOT (bmhd! AND cmap! AND body!)
  read_chunk
WEND
CLOSE #1
fastsave(SCREEN(1))
fastload
DO
LOOP
~MFREE(grafbuf%,120)
~MFREE(bitmapheader%,26)
END
'
PROCEDURE read_chunk
  BGET #1,grafbuf%,8
  len%={ADD(grafbuf%,4)}
  OPENW #0
  PRINT
  PRINT "chunk ";MKL$({grafbuf%});" länge ";
        len%;" Byte";
  SELECT MKL$({grafbuf%})
  CASE "BMHD"
    bitmapheader%=MALLOC(len%+6,&H10000)
    IF bitmapheader%
      read_bmhd              ! Bit-Map Header laden
    ELSE
      bmhd!=TRUE
      cmap!=TRUE
      body!=TRUE
    ENDIF
  CASE "CMAP"                ! Color Map laden
    read_cmap
    OPENS 1,0,0,CARD{ADD(bitmapheader%,16)},
          CARD{ADD(bitmapheader%,18)},
          BYTE{ADD(bitmapheader%,8)},
          CARD{ADD(bitmapheader%,20)}
    set_colors(SCREEN(1))
    IF CARD{ADD(bitmapheader%,18)} >
       CARD{ADD(bitmapheader%,2)}
      CARD{ADD(bitmapheader%,18)} =
         CARD{ADD(bitmapheader%,2)}
    ENDIF
  CASE "CAMG"                ! Screen-ViewMode
    BGET #1,grafbuf%,len%
    CARD{ADD(bitmapheader%,20)}={grafbuf%}
    PRINT " ";CARD{ADD(bitmapheader%,20)};
          " Screen Viewmode";
  CASE "BODY"
    read_body                ! Bild laden
  DEFAULT
    RELSEEK #1,len%-ODD(len%) ! überspringen
  ENDSELECT
RETURN
'
PROCEDURE read_bmhd
  bmhd!=TRUE
  BGET #1,bitmapheader%,len%
  PRINT
  PRINT "                  BITMAPHEADER"
  PRINT
  PRINT "Bitplane Weite          ";
        CARD{bitmapheader%}
  PRINT "Bitplane Höhe           ";
        CARD{bitmapheader%+2}
  PRINT "X-Offset in der Bitmap  ";
        CARD{bitmapheader%+4}
  PRINT "Y-Offset in der Bitmap  ";
        CARD{bitmapheader%+6}
  PRINT "Bitplane Tiefe          ";
        BYTE{bitmapheader%+8}
  PRINT "Maskierart              ";
        BYTE{bitmapheader%+9}
  PRINT "Komprimiert ja/nein     ";
        BYTE{bitmapheader%+10}
  PRINT "Maskierfarbe            ";
        CARD{bitmapheader%+12}
  PRINT "X-Seitenverhältnis      ";
        BYTE{bitmapheader%+14}
  PRINT "Y-Seitenverhältnis      ";
        BYTE{bitmapheader%+15}
  PRINT "Screen Weite            ";
        CARD{bitmapheader%+16}
  PRINT "Screen Höhe             ";
        CARD{bitmapheader%+18}
  PRINT
  sourcebytes&=SHR(CARD{bitmapheader%},3)
  destbytes&=SHR(CARD{ADD(bitmapheader%,16)},3)
  IF sourcebytes&<>destbytes&
    INC sourcebytes&
  ENDIF
RETURN
'
PROCEDURE read_cmap
  LOCAL colmap%,viewmod%
  cmap!=TRUE
  colmap%=MALLOC(len%+2,&H10000)
  IF colmap%
    CARD{colmap%}=len%
    BGET #1,ADD(colmap%,2),len%
  ELSE
    RELSEEK #1,len%-ODD(len%)
  ENDIF
  viewmod%=CARD{ADD(bitmapheader%,20)}
  IF viewmod%=0
    IF BYTE{ADD(bitmapheader%,8)}>5
      viewmod%=128
    ENDIF
    IF CARD{ADD(bitmapheader%,16)}>320
      viewmod%=viewmod% OR &H8000
    ENDIF
    IF CARD{ADD(bitmapheader%,18)}>256
```

→

```
        viewmod%=viewmod% OR &H4
      ENDIF
      CARD{ADD(bitmapheader%,20)}=viewmod%
    ENDIF
    {ADD(bitmapheader%,22)}=colmap%
  RETURN
  '
  PROCEDURE read_body ! schneller geht's nicht
    LOCAL hohe%,tiefe%,i%,code|,gen&
    body!=TRUE
    BACKS 1
    hohe&=CARD{ADD(bitmapheader%,18)}
    tiefe|=BYTE{ADD(bitmapheader%,8)}
    bitmap%=ADD(SCREEN(1),184)
    FOR i%=0 TO tiefe|
      bitplanes%(i%)={ADD(bitmap%,8+(i%*4))}
    NEXT i%
    IF 0=BYTE{ADD(bitmapheader%,10)}
      FOR rr&=0 TO PRED(hohe&)
        FOR pp&=0 TO PRED(tiefe|)
          i%=ADD(bitplanes%(pp&),
                 MUL(destbytes&,rr&))
          BGET #1,i%,sourcebytes&
        NEXT pp&
      NEXT rr&
    ELSE IF 1=BYTE{ADD(bitmapheader%,10)}
      FOR rr&=0 TO PRED(hohe&)   ! ByteRun1
        FOR pp&=0 TO PRED(tiefe|)
          i%=ADD(bitplanes%(pp&),
                 MUL(destbytes&,rr&))
          gen&=0
          WHILE gen&<sourcebytes&
            BGET #1,grafbuf%,1
            code|=BYTE{grafbuf%}
            IF code|<128
              BGET #1,ADD(i%,gen&),SUCC(code|)
              ADD gen&,SUCC(code|)
            ELSE IF code|>128
              BGET #1,grafbuf%,1
              byte|=BYTE{grafbuf%}
              FOR kk&=gen& TO
                  ADD(gen&,SUB(257,code|))
                BYTE{i%+kk&}=byte|
              NEXT kk&
              ADD gen&,SUB(257,code|)
            ENDIF
          WEND
        NEXT pp&
      NEXT rr&
    ENDIF
    FRONTS 1
  RETURN
  '
  PROCEDURE set_colors(scr%)
    LOCAL i%,colmap%,cols%,col%
    colmap%={ADD(bitmapheader%,22)}
    cols%=(CARD{colmap%}/3)
    colmap%=ADD(colmap%,2)
    FOR i%=0 TO cols%
      col%=ADD(colmap%,MUL(3,i%))
      ~SetRGB4(ADD(scr%,44),i%,
        SHR(BYTE{col%},4),
        SHR(BYTE{ADD(col%,1)},4),
        SHR(BYTE{ADD(col%,2)},4))
    NEXT i%
  RETURN
  '
  PROCEDURE fastsave(scr%)
    LOCAL tiefe&,farbanz&,farben%,farb$,
          bitmap%,bitplane%
    OPENS 0
    FRONTS 0
    FILESELECT "Fastpic speichern","speichern",
               "",file$
    IF LEN(file$)
      OPEN "O",#1,file$
      PRINT #1,"KICK";
      farbanz&=SHL(1,BYTE{ADD(scr%,189)})
      PRINT #1,MKI$(CARD{ADD(scr%,12)});!Weite
      PRINT #1,MKI$(CARD{ADD(scr%,14)});!Höhe
      tiefe&=BYTE{ADD(scr%,189)}
      PRINT #1,MKI$(tiefe&);! Tiefe
      PRINT #1,MKL$(CARD{ADD(scr%,76)});!ViewMode
      PRINT #1,MKI$(farbanz&);          !Farben
      farben%={ADD({ADD(scr%,48)},4)}
      farb$=SPACE$(MUL(farbanz&,2))
      BMOVE farben%,VARPTR(farb$),
            MUL(farbanz&,2)
      PRINT #1,farb$;
      bitmap%=ADD(scr%,184)
      FOR i%=0 TO PRED(BYTE{ADD(scr%,189)})
        bitplane%={ADD(bitmap%,
                   ADD(8,(MUL(i%,4))))}
        BPUT #1,bitplane%,
             MUL(DIV(CARD{ADD(scr%,12)},8),
             CARD{ADD(scr%,14)})
      NEXT i%
      CLOSE #1
    ENDIF
  RETURN
  '
  PROCEDURE fastload
    LOCAL farbanzahl&,farben$,bitmap%,bitplane%
    OPENS 0
    FRONTS 0
    FILESELECT "Fastpic laden","laden","",file$
    IF LEN(file$)
      OPEN "I",#1,file$
      test$=INPUT$(4,#1)
      IF test$<>"KICK"
        PRINT "Kein KICK Bild !"
        END
      ENDIF
      '
      scrbreite&=CVI(INPUT$(2,#1))
      scrhoehe&=CVI(INPUT$(2,#1))
      scrtiefe&=CVI(INPUT$(2,#1))
      modus%=CVL(INPUT$(4,#1))
      farbanzahl&=CVI(INPUT$(2,#1))
      OPENS 2,0,0,scrbreite&,scrhoehe&,
            scrtiefe&,modus%
      bitmap%=ADD(SCREEN(2),184)
      DISPLAY OFF
      farben$=INPUT$(MUL(farbanzahl&,2),#1)
      ~LoadRGB4(ADD(SCREEN(2),44),
                VARPTR(farben$),farbanzahl&)
      FOR i%=0 TO PRED(scrtiefe&)
        bitplane%={ADD(bitmap%,
                   ADD(8,(MUL(i%,4))))}
        BGET #1,bitplane%,MUL(DIV(scrbreite&,8),
                scrhoehe&)
      NEXT i%
      DISPLAY ON
      CLOSE #1
    ENDIF
  RETURN
```
□

Jetzt für AMIGA

# PICTURE-DISKS GRAFIKSAMMLUNG

Die Picture-Disks-Serie ist die erste Graphik-Sammlung mit hochauflösenden Graphiken für Commodore Amiga Computer. Die Picture-Disks sind von der Qualität einzigartig und decken alle Themenbereiche ab.

1000 Graphiken DM 98,—
2000 Graphiken DM 189,—

Die hochauflösenden Graphiken im IFF-Format können ohne Probleme in allen gängigen Graphik-, Mal- oder DTP-Programme der AMIGA Welt übernommen und dort weiterverarbeitet werden. Ein schnelles Auffinden der einzelnen Graphiken garantiert ein mitgeliefertes Handbuch. Picture-Disks sind ein Muß für alle professionellen DTP-, Mal- und Textverarbeitungssysteme mit Graphikeinbindung.

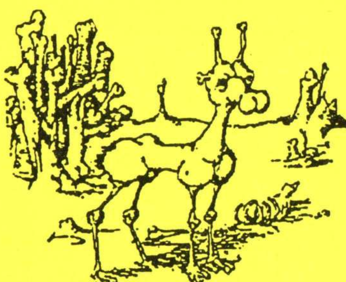
Unverbindlich empfohlene Verkaufspreise





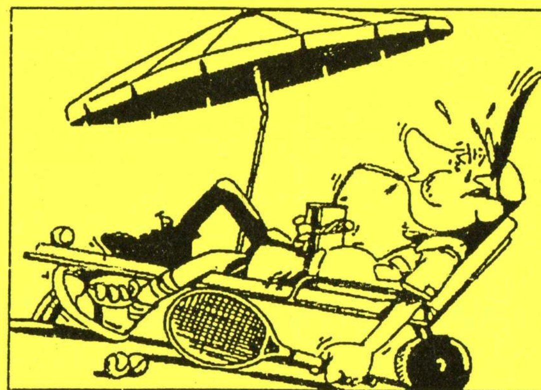














PICTURE DISKS erhalten Sie im guten Fachhandel oder direkt beim Heim-Verlag

**Heim Verlag**

*Heidelberger Landstr. 194*
*6100 Darmstadt-Eberstadt*
*Telefon 06151-56057*

Schweiz: DataTrade AG
Langstr. 94
CH-8021 Zürich

## BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle ☐ 1000 AMIGA PICTURE-DISKS zum Preis von DM 98,—
☐ 2000 AMIGA PICTURE-DISKS zum Preis von DM 189,—
zzgl. DM 5,– Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ______________________

Straße, Hausnr. ______________________

PLZ, Ort ______________________

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte.

# KICKS FÜR INSIDER

## Reine Zauberei

Der Umgang mit dem Computer mag einigen Leuten wie reine Zauberei vorkommen. Aber wie unser ferner Mitarbeiter hier demonstriert, ist alles mit einiger Übung und mit so manchem Trick zu verwirklichen. Einige dieser Tricks werden wir Ihnen in den folgenden KICKS präsentieren. Zwar wird sich der Rechner nicht gleich in die Lüfte erheben, jedoch kommt er ganz schön ins rotieren.

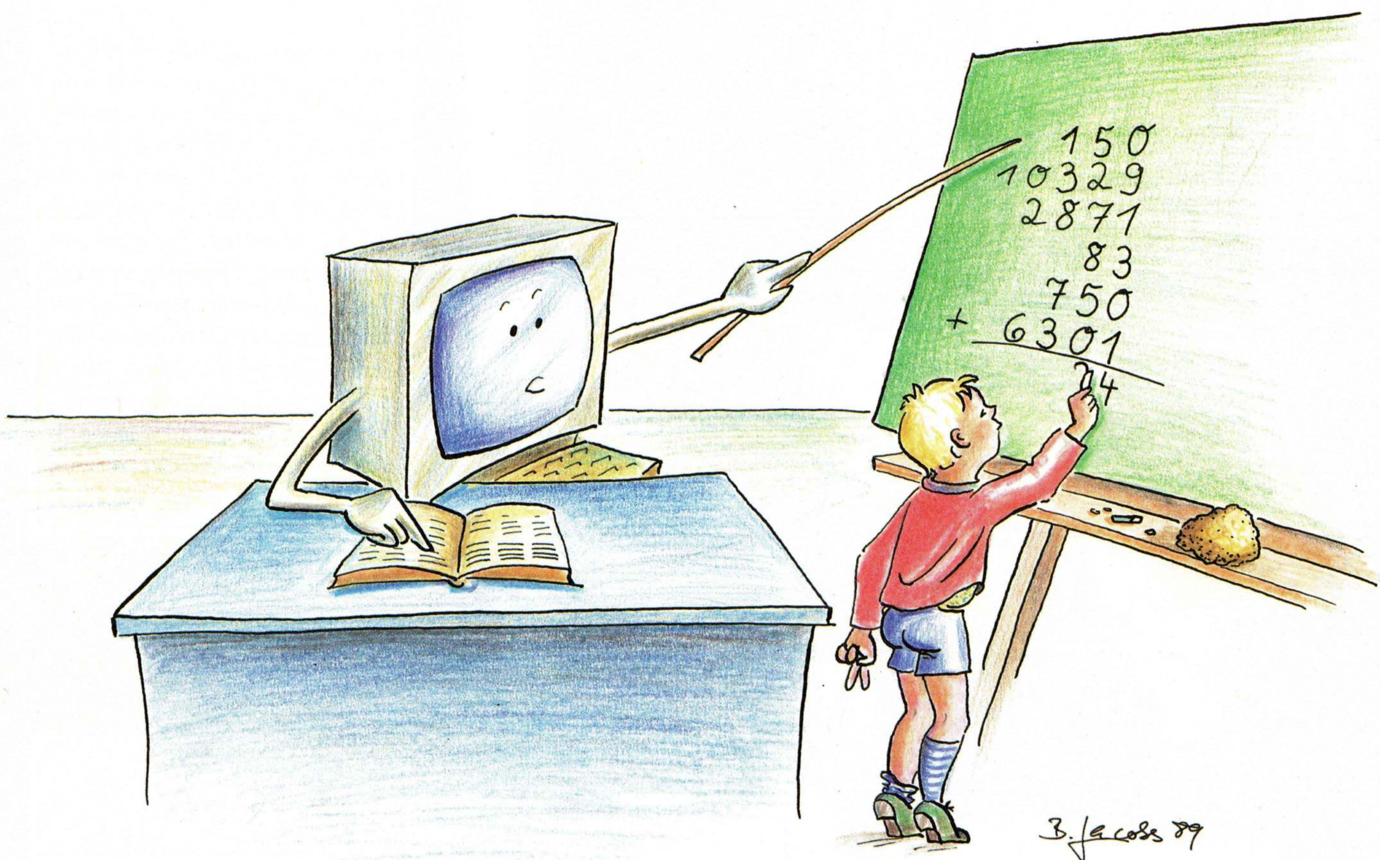
Die Red.

## INHALT

VON THOMAS HOHENBERGER

# KickSummer

## *Ohne Kontrolle geht es nicht*




*Wer kennt das nicht: Man liest in einer Zeitschrift, wie man die Kickstartsoftware des AMIGA 1000 endlich optimal auf seine Speichererweiterung anpassen kann, oder wie man das DOS-Fenster auf volle Größe bringt, oder wie man einfach seine eigene DOS-Meldung nach dem Reset auf den Bildschirm bekommt.*

Einfach einen Disk-Monitor laden, die Änderungen vornehmen, ohne Korrektur der Block-Checksumme die Daten zurück auf die Disk schreiben. Nun die Kickstart neu booten und schon funktioniert das neue Betriebssystem!? Aber beim ersten Reset stellt man fest, daß der AMIGA statt einer anderen Workbench wieder die Kickstartdiskette verlangt. Das liegt daran, daß beim Reset eine Prüfsumme über das Betriebssystem-WOM (Write once Memory) gebildet wird, und falls diese nicht korrekt ist wird die Kickstartsoftware neu geladen. Die Berechnung die-

ser Prüfsumme ist nicht so einfach ohne geeignetes Programm durchzuführen. Hier jedoch greift der KickSummer ein. Er liest das Betriebssystem von einer Kickstartdisk in DF0:, berechnet die Prüfsumme und korrigiert diese, wenn nötig. Der Benutzer muß lediglich das Programm vom CLI aus aufrufen, die nicht schreibgeschützte Kickstartdisk einlegen und die Sicherheitsabfrage mit "ja" (ich will) beantworten. Alles andere übernimmt das Programm.

"KickSummer" ist vollständig in Maschinensprache geschrieben. Hierbei wurde der KUMA-Seka-Assembler verwendet. Bitte geben Sie das Listing mit den Editor des Assemblers ein und assemblieren (a) Sie ohne Optionen. Das fertige Programm können Sie dann auf Disk schreiben (wo). Es sollte jedoch auch problemlos mit anderen Assemblern wie dem Profimat zu erzeugen sein.

Das fertige Programm ist nur gute zwei Kilobytes lang, so daß es auf Diskette nur 5 Blocks belegt. Es kann also auf der Workbench in den c-Ordner oder auf einer anderen Arbeitsdiskette abgelegt werden.

Hat man eine Kickstart modifiziert, muß man sich nicht mehr um die Prüfsumme kümmern, sondern braucht vor deren Gebrauch nur noch den Checksummer aufzurufen. Hoffen wir, daß die geänderte Systemsoftware auch wirklich noch funktioniert.

## *Zur Bedienung:*

Das Programm wird mit "KickSummer" vom CLI aus aufgerufen. Nun muß die Kickstartdisk in Drive 0 eingelegt werden. Wird die folgende Sicherheitsabfrage mit "nein" beantwortet, wird das Programm verlassen. Ansonsten muß mit ja geantwortet werden. Alles andere erfolgt automatisch. Ist keine Kickstartdisk eingelegt worden oder nicht genügend Speicher vorhanden, bricht das Programm mit einer entsprechenden Meldung ab. Da das Programm 256 Kilobytes Chip-RAM belegt, um die gesamte Kickstart laden zu können, sollten bei Speichermangel Fenster und Screens geschlossen oder andere Programme zuerst beendet werden. Ist die Prüfsumme bereits in Ordnung, endet das Programm ebenfalls mit der Ausgabe der Summe. Anderenfalls wird die Prüfsumme korrigiert und zurückgeschrieben. Alte und neue Prüfsumme werden angezeigt, und das Programm endet. Ab jetzt sollte die Kickstart problemlos funktionieren.

```
; KickSummer
; Korrigiert die Checksumme einer modifizierten Kickstartdisk
; Autor: Thomas Hohenberger
; (c) MAXON Computer GmbH 1989
; KICKSTART 1989

ExecBase = 4

; Exec Offsets:

FindTask = -294
AddPort = -354
RemPort = -360
OpenDev = -444
CloseDev = -450
DoIO = -456
AllocMem = -198
FreeMem = -210
OpenLib = -552
CloseLib = -414

; Dos Offsets:

Read = -42
Write = -48
Input = -54
Output = -60

; Konstanten:

len = $40000
Chip = 3
kick = $4b49434b
ja = 'j'*$100+'a'
nein1 = 'n'*$100+'e'
nein2 = 'i'*$100+'n'

; Programm:

run:            move.l ExecBase,a6
                clr.l d0
                lea DosName,a1
                jsr OpenLib(a6)
                move.l d0,DosBase
                tst.l d0
                bne Cont1
                rts

Cont1:          move.l d0,a6
                jsr Output(a6)
                move.l d0,OutHandle
                move.l DosBase,a6
                jsr Input(a6)
                move.l d0,InHandle
                move.l OutHandle,d1
                move.l
#Itext,d2
                move.l #Itlen,d3
                move.l DosBase,a6
                jsr Write(a6)

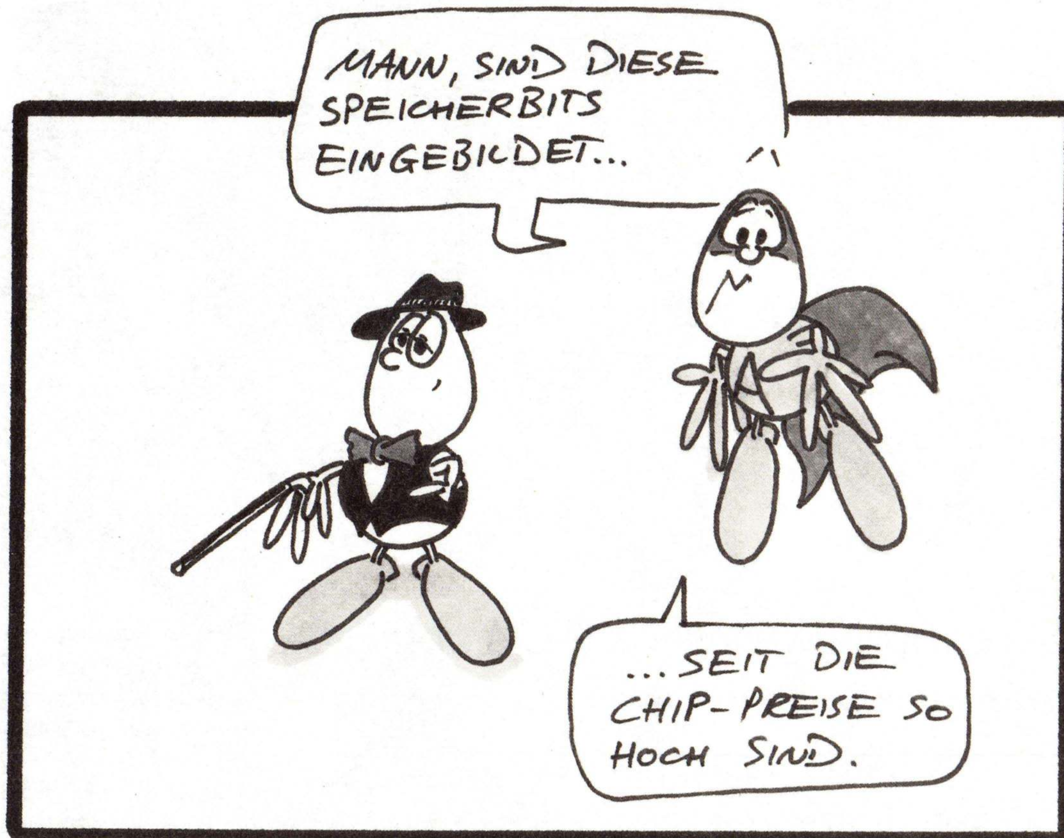
il:             move.l OutHandle,d1
                move.l #JNtext,d2
                move.l #JNtlen,d3
                move.l DosBase,a6
                jsr Write(a6)
                move.l InHandle,d1
                move.l #InPuffer,d2
                move.l #Inlen,d3
                move.l DosBase,a6
                jsr Read(a6)
                cmp.w #ja,InPuffer
                beq Cont2
                cmp.w #nein1,InPuffer
                bne il
                cmp.w #nein2,InPuffer+2
                bne il
```



```
                move.l OutHandle,d1
                move.l #Atext,d2
                move.l #Atlen,d3
                move.l DosBase,a6
                jsr Write(a6)
                bra CloseDos
Cont2:          move.l #len,d0
                move.l #Chip,d1
                move.l ExecBase,a6
                jsr AllocMem(a6)
                tst.l d0
                bne Cont3
                move.l OutHandle,d1
                move.l #Mtext,d2
                move.l #Mtlen,d3
                move.l DosBase,a6
                jsr Write(a6)
                bra CloseDos
Cont3:          move.l d0,Buffer
                move.l #512,DIOlen
                move.l #0,DIOoffs
                bsr DRead
                move.l Buffer,a0
                cmp.l #kick,(a0)
                beq Cont4
                move.l OutHandle,d1
                move.l #Ktext,d2
                move.l #Ktlen,d3
                move.l DosBase,a6
                jsr Write(a6)
                bra Exit
Cont4:          move.l #len,DIOlen
                move.l #512,DIOoffs
                bsr DRead
                move.l Buffer,a0
                move.l a0,a1
                add.l #$40000,a1
                bsr Sum
                move.l d0,Summe
                tst.l d0
                bne Cont5
                move.l Buffer,a0
                add.l #$3ffe8,a0
                move.l (a0),d0
                move.l d0,Summe
                move.l d0,OldS
                bsr DispSums
                move.l OutHandle,d1
                move.l #Otext,d2
                move.l #Otlen,d3
                move.l DosBase,a6
                jsr Write(a6)
                bra  Exit
Cont5:          move.l #$40000,DIOoffs
                move.l #512,DIOlen
                bsr DRead
                move.l Buffer,a0
                add.l #$1e8,a0
                move.l (a0),d0
                move.l d0,OldS
                sub.l Summe,d0
                move.l d0,(a0)
                move.l d0,Summe
                move.l #$40000,DIOoffs
                move.l #512,DIOlen
                bsr DWrite
                bsr DispSums
                move.l OutHandle,d1
                move.l #FText,d2
                move.l #Ftlen,d3
                move.l DosBase,a6
                jsr Write(a6)
Exit:           move.l Buffer,a1
                move.l #len,d0
                move.l ExecBase,a6
                jsr FreeMem(a6)
CloseDos:       move.l DosBase,a1
                move.l ExecBase,a6
                jsr CloseLib(a6)
Error:          rts
DispSums:       move.l OldS,d0
                jsr Disp
                move.l newtxt,stxtc
                move.l Summe,d0
                jsr Disp
                rts
Disp:           move.l d0,DTemp
                lea DTemp,a0
                lea ascs,a1
                lea hex,a2
                clr.l d0
                move.l #3,d1
hl:             move.b (a0)+,d0
                move.b d0,d7
                and.l #$f0,d0
                lsr.b #4,d0
                move.b (a2,d0),d6
                move.b d6,(a1)+
                move.b d7,d0
                and.l #$0f,d0
                move.b (a2,d0),d6
                move.b d6,(a1)+
                dbra d1,hl
                move.l OutHandle,d1
                move.l #stxt,d2
                move.l #stxlen,d3
                move.l DosBase,a6
                jsr Write(a6)
                rts
Sum:            clr.l d0              ;a1 Bereich
sumloop:        move.l (a0),d1        ;d0 Ergebnis
                add.l d0,d1
                cmp.l d0,d1
                bge sumskip
                add.l #1,d0
sumskip:        add.l (a0)+,d0
                cmp.l a1,a0
                bcs sumloop
                rts
DRead:          move.w #2,Mode
                bra Access
DWrite:         move.w #3,Mode
Access:         move.l ExecBase,a6
                lea 0,a1
                jsr FindTask(a6)
                move.l d0,Port+$10
                lea Port,a1
                jsr AddPort(a6)
                lea DIO,a1
                clr.l d0
                clr.l d1
                lea tddname,a0
                jsr OpenDev(a6)
                tst.l d0
                bne Exit
                lea DIO,a1
                move.l #Port,14(a1)
                move.w Mode,28(a1)
                move.l Buffer,40(a1)
                move.l DIOlen,36(a1)
                move.l DIOoffs,44(a1)
                move.l ExecBase,a6
                jsr DoIO(a6)
                move.l DIO+32,d6
                lea DIO,a1
                move.w #9,28(a1)
                move.l #0,36(a1)
                jsr DoIO(a6)
                lea Port,a1
                jsr RemPort(a6)
                lea DIO,a1
                jsr CloseDev(a6)
                rts
                even
```

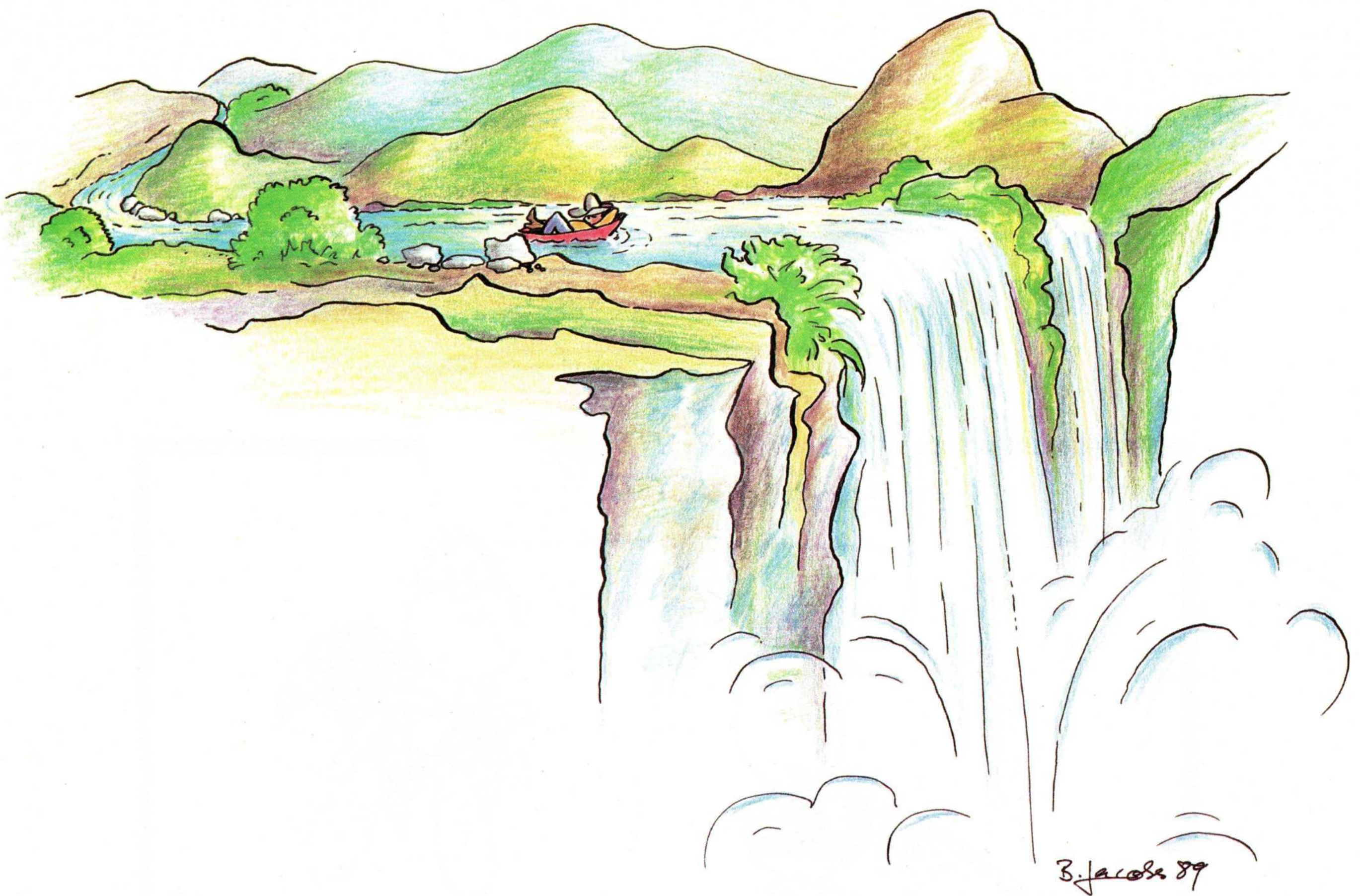


```
tddname:        dc.b 'trackdisk.device',0
                even
Mode:           dc.w 0
                even
dosname:        dc.b 'dos.library',0
                even
DosBase:        dc.l 0
OutHandle:      dc.l 0
InHandle:       dc.l 0
Buffer:         dc.l 0
Port:           blk.l 8,0
DIO:            blk.l 20,0
DIOlen:         dc.l 0
DIOoffs:        dc.l 0
Summe:          dc.l 0
OldS:           dc.l 0
DTemp:          dc.l 0
        even
Itext:  dc.b 10
        dc.b 'KickSummer: Written April 1988
           by Thomas Hohenberger'
        dc.b 10,10
        dc.b 'Es wird die Checksumme einer
           modifizierten Kickstart'
        dc.b 'diskette neu',10
        dc.b 'berechnet und korrigiert.',10,10
        dc.b 'Bitte Kickstartdisk in DF0:
           einlegen!',10,10
Itlen = * - Itext
        even
JNtext: dc.b 'Soll die Kickstartchecksumme
           korrigiert werden '
        dc.b '(ja/nein)? '
JNtlen = * - JNtext
        even
Atext:  dc.b 10,10
        dc.b 'Programm abgebrochen!',10,10
Atlen = * - Atext
        even
Mtext:  dc.b 10,10
        dc.b 'Nicht genügend Speicher
           vorhanden!',10,10
Mtlen = * - - Mtext
        even
Ktext:  dc.b 10,10
        dc.b 'Keine KickstartDisk
           eingelegt!',10,10
Ktlen = * - Ktext
        even
Otext:  dc.b 10,10
        dc.b 'Checksumme bereits k!',10,10
Otlen = * - Otext
        even
Ftext:  dc.b 10,10
        dc.b 'Checksumme korrigiert!',10
        dc.b 'Die KickstartDisk sollte nun
           ordentlich Arbeiten.'
        dc.b 10,10
Ftlen = * - Ftext
        even
newtxt: dc.b 'Neue'
        odd
stxt:   dc.b 10
stxtc:  dc.b 'Alte Checksumme: $'
ascs:   dc.b '00000000'
stxlen = * - - stxt
        even
hex:    dc.b '0123456789abcdef'
        even
InPuffer:
        blk.b 80,0
Inlen = * - InPuffer
;Ende Listing
```



VON DANIEL GEMBRIS

# Don't worry, be happy!




*Hassen Sie es auch, wenn Ihr AMIGA abstürzt und ohne Vorankündigung eine Gurumeditation erscheint? Dann brauchen Sie das Programm "Don't worry, be happy!", das Sie akustisch (musikalisch) auf eine Gurumeditation einstimmt!*

Das Programm ist in Assembler geschrieben und resetfest. Der Sound, der vor jedem Guru abgespielt werden soll, muß ab dem Label "dt" abgelegt sein. Um Sounds, die im 8SVX-IF-Format vorliegen, einzubinden, kann das abgedruckte Basic-Programm 8SVXtoAss verwendet werden. Das Programm schneidet die Informationen, die vor den Musikdaten stehen, ab, wandelt die Daten in das ASCII-

Format um und versieht jede Datenzeile mit der Assembler-Anweisung “dc.b”. Weil in keinem der beiden Programme ein Entpacker integriert ist, müssen die Musikdaten ungepackt vorliegen. Die vorliegende Sound-ASCII-Datei kann dann in das Hauptprogramm eingebunden werden.

Die Lautstärke des Sounds wird am Anfang des Listings in der Variablen “vol” definiert. Die Variable “rate” bestimmt sinnvollerweise die Abspielrate.

Der Sound wird von dem Custom-Chip “Paula” im Loop-Modus abgespielt, wenn die Sound-DMA-Kanäle einmal eingeschaltet wurden. D.h. der Sound beginnt immer wieder von vorne. Zu stoppen ist die Ausgabe nur durch das Abschalten der DMA-Kanäle. Dieses sollte aber möglich ist nicht geschehen, wenn die Musik o.ä. noch nicht zu Ende ist. Da die Paula nicht “riechen” kann, wann die Soundausgabe beendet werden soll, muß eine Warteschleife benutzt werden. Dazu müssen die Spieldauer des Sounds mit einer Stoppuhr ermittelt und die gemessene Zeit in Sekunden in das Listing eingetragen werden (Variable: “dauer”). Natürlich kann es auch durch Probieren angepaßt werden.

Am Ende der Datenliste müssen die Labels “dtend”,”soundE” und “Ende” ergänzt werden, wenn man beabsichtigt, das Programm und die Sounddaten mit dem CLI-Befehl “JOIN” zusammenzusetzen. Dadurch, daß die Musik o.ä. nicht nachgeladen werden muß, ist erstens der Überraschungseffekt viel ausgeprägter und zweitens braucht man nicht immer nach der Diskette zu suchen, auf der sich der Sound befindet.

Die Soundausgabe erfolgt bei dem Programm “Don’t worry, be happy!” nur vor Gurus, die durch eine Prozessor-Exception ausgelöst wurden.

Bei “Don’t worry be happy!” wird, wenn versucht wird das Programm zu starten, obwohl die Prozessor-Exception-Vektoren schon umgebogen sind, das Programm vorzeitig abgebrochen und eine kleine Meldung ausgegeben, die den User darauf hinweist, daß das Programm bereits installiert ist.

Ach so, raten Sie mal, warum das Programm so heißt, wie es heißt! Na? Ist doch klar, das Lied “Don’t worry, be happy” paßt doch gut vor eine Gurumeditation, oder?!

Das Programm ist mit dem KICK-ASS-Assembler geschrieben, der SEKA-Assembler kann aber auch herangezogen werden. Aus Platzgründen und um Ihnen die Tipparbeit von endlosen Datas zu ersparen, gibt es das abgedruckte Programm vor einem Guru nur einen Ton aus. Auf unserer Service-Diskette finden Sie aber das richtige ‘Dont’t worry, be happy’-Programm. Natürlich steht Ihnen offen andere Sounds zu verwenden, die Sie vielleicht selber digitalisiert haben. Bedenken Sie aber, daß die Sounds ChipMEM benötigen, und, da das Programm resetfest ist, es erbarmungslos den Speicher anfordert. Am besten rufen Sie das Programm in der startup-sequence auf, so ist es sofort installiert.

Neben dem eigentlichen Hauptprogramm finden Sie das kleine AMIGA-BASIC-Programm, das die Assembler-Daten erzeugt.

```
REM 8SVX to Ass
REM (w) by Daniel Gembris
REM
REM  Mit diesem Programm lassen sich digitalisierte Musikstücke
REM  in Assembler-"dc.b-Wüsten" umwandeln. Dadurch brauchen
REM  Musikstücke nun nicht mehr nachgeladen werden.

CLS
INPUT "Name des umzuwandelnden Musikstücks";Filename$
INPUT "Unter welchem Namen soll es abgespeichert werden";ou$
OPEN "I",1,Filename$
OPEN "O",2,ou$
WHILE a$<>"B"
 a$=INPUT$(1,#1)
WEND
WHILE a$<>"O"
 a$=INPUT$(1,#1)
WEND
WHILE a$<>"D"
 a$=INPUT$(1,#1)
WEND
a$=INPUT$(1,#1)
WHILE NOT EOF(1)
 a$=" dc.b "
 WHILE LEN(a$)<65
  IF NOT EOF(1) THEN h=ASC(INPUT$(1,#1))
  a$=a$+RIGHT$(STR$(h),(LEN(STR$(h))-1))
  IF LEN(a$)<65 THEN a$=a$+","
 WEND
 IF RIGHT$(a$,1)="," THEN a$=LEFT$(a$,(LEN(a$)-1))
 PRINT #2,a$
WEND
CLOSE(1)
CLOSE(2)
```

```
;Don't worry be happy!
;Autor: Daniel Gembris
;(c) MAXON Computer GmbH
;KICKSTART 1989

execbase = 4
AllocMem = -$C6
AllocAbs = -204
freemem = -210
OpenLib =  -408
CloseLib = -414
require = 2
CoolCapture = 46
ColdCapture = 42
DisplayAlert = -90
toffset0 = buserror-sound
toffset1 = addresserror-sound
toffset2 = illegalop-sound
toffset3 = divisionbyz-sound
toffset5 = aopex-sound
```
→

```
toffset6 = fopex-sound
toffset7 = uninitinter-sound
toffset8 = unpermittedinter-sound
laenge = dtend-dt
rate = 150
vol = 64
dauer = 90

   move.l  $8.s,d0 ;== testen, ob Programm bereits
   and.l  #$00F00000,d0   ;== installiert
   bne  notyet            ;noch nicht
   move.l  execbase.s,a6  ;Intuition-Library öffnen
   lea  i_name,a1
   moveq  #0,d0
   jsr  OpenLib(a6)
   move.l  d0,i_base
   move.l  d0,a6
   beq  Ende0
   moveq  #0,d0           ;Alert ausgeben
   lea  string,a0
   moveq  #48,d1
   jsr  DisplayAlert(a6)
   move.l  execbase.s,a6  ;Library schließen
   move.l  i_base,a1
   jsr  CloseLib(a6)
Ende0: rts
notyet:
   move.l  execbase.s,a6  ;== Install-Routine (IR) für
   move.l  #Ende-Anfang,d0 ;== neuen "Alert" resetfest machen
   moveq  #require,d1
   jsr  AllocMem(a6)  ;Speicher für IR anfor.
   tst.l  d0
   beq  fehler            ;hat nicht geklappt
   move.l  d0,a1
   move.l  a1,CoolCapture(a6) ;CoolCapture-Vektor auf IR
   clr.l  ColdCapture(a6) ;ColdCapture-Vektor nicht nötig
   move.w  #Ende-Anfang,d0    ;IR kopieren
   lea  Anfang,a0
loop1: move.b  (a0)+,(a1)+
   dbra  d0,loop1
   jsr  Prüfsumme     ;Prüfsumme neu berechnen
fehler: movem.l  d0-d7/a0-a6,-(sp) ;Register retten
   jmp weiter

Anfang: ;=====Installationsroutine======
   movem.l  d0-d7/a0-a6,-(sp) ;Register retten
   move.l  execbase.s,a6      ;IR vor Löschung schützen
   move.l  #Ende-Anfang,d0
   lea  Anfang(pc),a1
   jsr  AllocAbs(a6)
   lea  Anfang(pc),a1      ;CoolCapture-Vektor neu setzen
   move.l  a1,CoolCapture(a6)
   clr.l  ColdCapture(a6)
weiter:
   move.l  #soundE-sound,d0  ;Speicher für neuen "Alert"
   move.l  execbase.s,a6
   moveq  #require,d1
   jsr  AllocMem(a6)
   lea  adresse(pc),a0 ;Adresse merken
   move.l  d0,(a0)
   beq  error           ;kein Speicher verfügbar

   lea  adresse(pc),a2 ;Trap-Vektoren auf Debugger
   move.l  (a2),a2
   lea  TrapData(pc),a1 ;ab TrapData werden Adressen
   ;auf die die Trap-Pointer zeigen, abgelegt
   lea  Traps(pc),a3   ;ab Traps stehen Trap-Adressen
   lea  TrapOffsets(pc),a4 ;ab TrapOffsets stehen Offsets
   move.w  #7,d0  ;sie werden zu sound addiert
```

```
loop5: move.l  (a3)+,a5;für neue Trap-Routinen
   move.l  (a5),(a1)+
   move.l  (a4)+,d1
   lea  0(a2,d1),a0
   move.l  a0,(a5)
   dbra  d0,loop5

   lea  adresse(pc),a1      ;Routine kopieren
   move.l  (a1),a1
   move.l  #soundE-sound-1,d1
   lea  sound(pc),a0
loop3: move.b  (a0)+,(a1)+
   dbra  d1,loop3
error: movem.l  (sp)+,d0-d7/a0-a6 ;Register mit alten Werten
   rts                          ;beschreiben
adresse: dc.l 0

sound: jmp  sound1(pc)

Prüfsumme: ;Prüfsumme neu berechnen
   move.l  execbase.s,a6
   clr.l  d1
   lea  34(a6),a0
   move.w  #$16,d0
loop2: add.w  (a0)+,d1
   dbra  d0,loop2
   not.w  d1
   move.w  d1,82(a6)
   rts

buserror: movem.l  d0-d7/a0-a7,-(sp)
   moveq  #$0,d0
   jmp  sound(pc)
addresserror: movem.l  d0-d7/a0-a7,-(sp)
   moveq  #4,d0
   jmp  sound(pc)
illegalop: movem.l  d0-d7/a0-a7,-(sp)
   moveq  #8,d0
   jmp  sound(pc)
divisionbyz: movem.l  d0-d7/a0-a7,-(sp)
   moveq  #12,d0
   jmp  sound(pc)
aopex: movem.l  d0-d7/a0-a7,-(sp)
   moveq  #16,d0
   jmp  sound(pc)
fopex: movem.l  d0-d7/a0-a7,-(sp)
   moveq  #20,d0
   jmp  sound(pc)
uninitinter: movem.l  d0-d7/a0-a7,-(sp)
   moveq  #24,d0
   jmp  sound(pc)
unpermittedinter: movem.l  d0-d7/a0-a7,-(sp)
   moveq  #28,d0
   jmp  sound(pc)

radr: dc.l 0
tnumber: dc.l $FFFFFFFF
save0: dc.l 0

string:
   dc.b 0,66,16,' GURUMUSIK bereits installiert !!',0,255
   dc.b 0,66,26,' OK                  OK',0,0

i_base: dc.l 0
i_name: dc.b 'intuition.library',0,0
   even

Traps:
   dc.l $000008,$00000C,$000010,$000014,$000028,
        $00002C,$00003C
   dc.l  $000060

TrapData:
   blk.l  8

TrapOffsets:
   dc.l  toffset0,toffset1,toffset2,toffset3,
         toffset5,toffset6
   dc.l  toffset7,toffset8

sound1: ;der neue "Alert":                    →
```

```
    lea  tnumber(pc),a0    ;Trap-Nummer merken
    move.l  d0,(a0)

    move.l execbase.s,a6   ;Sp.für Sound-Daten
    moveq #laenge,d0       ;anfordern
    moveq  #2,d1
    jsr  AllocMem(a6)
    lea saddress(pc),a0    ;Adresse retten
    move.l  d0,(a0)
    beq  endx          ;keinen Speicher erhalten
    move.l  d0,a1      ;Sound-Daten kopieren
    lea  dt(pc),a0
    moveq  #laenge-1,d0
cloop: move.b  (a0)+,(a1)+
    dbra  d0,cloop

    lea  $DFF0A0,a0 ;Sound-DMA initialisieren
    lea  saddress(pc),a1
    move.l  (a1),(a0)+
    move.w  #laenge/2,(a0)+
    move.w  #rate,(a0)+
    move.w  #vol,(a0)
    move.w  #$8201,$DFF096  ;Kanal 0 aktivieren

    jsr  cnt(pc)       ;aktuellen Zählerstand des
    move.l  d0,d1      ;50Hz Timers ermitteln
    add.l  #dauer,d1 ;90/50 sec. Verzögerung
    and.l  #$FFFFFF,d1 ;möglichen Übertrag
                        ignorieren
delay: jsr  cnt(pc)  ;warten, bis errechneter
                       Stand
    cmp.l d0,d1        ;erreicht
    bne  delay

    move.w  #$F,$DFF096 ;alle Audio-Kanäle
                          abschalten
    lea  saddress(pc),a1
    move.l  (a1),a1
    moveq  #laenge,d0    ;Speicher an System
                          zurückgeben
    jsr  freemem(a6)
endx:
    lea  TrapData(pc),a0  ;Rücksprungadresse
                            ermitteln
    lea  tnumber(pc),a1
    move.l  (a1),d0
    lea  radr(pc),a2 ;Rücksprungadresse in radr
    move.l  0(a0,d0),(a2)
    movem.l  (sp)+,d0-d7/a0-a7
    move.l  radr(pc),-(sp)  ;in Trap-Routine
                              springen
    rts

saddress: dc.l 0
counter:
cn: dc.b 0
cmsb: dc.b 0
cmid: dc.b 0
clsb: dc.b 0

cnt:
    lea  clsb(pc),a0
    move.b  $BFE801,(a0)
    lea  cmid(pc),a0
    move.b  $BFE901,(a0)
    lea  cmsb(pc),a0
    move.b  $BFEA01,(a0)
    lea  counter(pc),a0
    move.l  (a0),d0
    rts

dt:;Es folgen Sounddaten, an dieser St.muß
    der gesampelte
   ;Sound eingesetzt werden. Diese Werte geben
    nur einen
   ;nervenden Ton aus.
dc.b 0,0,1,124,0,0,0,0,254,252,252,250,248,
     248,246,246,246,244,244
dc.b 240,242,244,244,246,250,252,252,254,0,
     2,4,8,14,14,18,22,22
dtend:
soundE:
Ende:
```

END of KICK

# KICKS FUR INSIDER

Liebe Leser, KICKS für INSIDER ist eine Rubrik, die für interessante und lehrreiche Programme geschaffen wurde, denn was nutzt das schönste Programm, wenn man es nur abtippen, aber nicht verstehen kann.

Um jeden Monat neue KICKS zu veröffentlichen, sind wir natürlich auch von unseren Lesern abhängig, denn irgendwann einmal sind die Einfälle jedes Programmierers erschöpft. Also, nehmen Sie an dieser Rubrik aktiv teil, ergänzen Sie sie mit Ihren eigenen Programmen. Werden Sie Autor in der KICKSTART.

Dabei sind alle auf dem AMIGA existierenden Programmiersprachen gewünscht. Egal, ob Basic, C, Assembler, Modula 2, Pascal oder andere.

## WIE DAS GEHT?

Ganz einfach. Schicken Sie uns das Programm, samt dem zugehörigen Artikel, eventuellen Bildern und Hardcopies (nach Möglichkeit "gegrabbt") auf Diskette zu. Wir werden uns dann bei Ihnen melden. Ein interessantes Honorar wartet bei Veröffentlichung natürlich auch auf Sie.

Wenn Sie Fragen haben, rufen Sie doch einfach bei uns an.

Tel. 06196/481811
Mo-Fr. 11 bis 17 Uhr

*Ihre KICKSTART-Redaktion*

# AZTEC-C FÜR AMIGA VERSION 3.6

Wußten Sie, daß eines der verbreitesten und komplexesten Betriebssysteme - UNIX - in C geschrieben ist ?

Wußten Sie, daß auch das Betriebssystem des AMIGA größtenteils in C geschrieben wurde ?

Wußten Sie, daß C eine der wichtigsten und modernsten Programmiersprachen ist ?

Wußten Sie, daß Aztec-C einer der schnellsten und leistungsfähigsten Compiler für den AMIGA ist ?

Wußten Sie, daß jetzt Aztec-C in der Version 3.6 verfügbar ist ?

Möchten Sie mehr darüber wissen? Dann schicken Sie uns einen ausreichend frankierten Rückumschlag und Sie erhalten ausführliche Information.

HIERMIT BESTELLE ICH:

- ☐ AZTEC-C68K/AM-P PROFESSIONAL SYSTEM FÜR DM 298,-
- ☐ AZTEC-C68K/AM-D DEVELOPER SYSTEM FÜR DM 598.-
- ☐ AZTEC-SDB SOURCE LEVEL DEBUGGER FÜR DM 149.-

Versandkosten: Inland DM 7,50 Ausland DM 10,-
Auslandbestellungen nur gegen Vorauskasse
Nachnahmegebühr DM 3,70

- ☐ Vorauskasse
- ☐ Nachnahme

NAME: ____________________

VORNAME: ____________________

STRAßE: ____________________

ORT: ____________________

UNTERSCHRIFT: ____________________

AZTEC-C IST EIN EINGETRAGENES WARENZEICHEN VON MANX SOFTWARE SYSTEMS




MAXON COMPUTER GMBH
INDUSTRIESTRAßE 26
6236 ESCHBORN
TEL. 06196/481811

VON MICHAEL TEISTLER

# Jetzt wird´s bunt!!!

## Reizvolle Änderung der Farben




*Nun gut, Sie haben eine IFF-Bilder-Sammlung mit über 100 Bildern... Aber ist denn damit alleine schon alles ausgereizt??? Kramen Sie einfach Ihre (verstaubten) Disketten mit den Bildern wieder hervor und benutzen Sie COLORPLAY - ein Programm zur reizvollen Veränderung der Farben eines beliebigen Bildes.*

Als Programmiersprache dient AMIGABASIC, ist also für jeden AMIGA-Freak zu benutzen.

Eine kleine Bemerkung, bevor es losgeht: Das Programm selbst ändert nur die Farben, enthält aber keinen IFF-Lader oder ähnliches. Deshalb empfiehlt es sich, "Colorplay" z.B. mit dem Programm "LoadIlbm-SaveAcbm" von der Extras-Diskette zu verbinden. Die Colorplay-Routinen (also eigentlich das gesamte Programm) müßten hier vor dem Label "mcleanup" eingesetzt wer-

den. Bei dieser Version müßte man dann nur noch den Namen des IFF-Bildes eingeben, und könnte danach alle Effekte daran ausprobieren.

Bei "Colorplay" werden alle Schritte, bei denen Farben verändert werden, per Mausklick 'ausgelöst'.

## Hier die Reihenfolge:

Zuerst wird das Ausgangsbild in mehreren Schritten in ein Schwarzweißbild umgewandelt (wenn es das vorher schon war, entsteht hier natürlich keine Veränderung). Diese SW-Darstellung erfolgt erst in 16 und dann in 8 bzw. 4 Graustufen. Mit weiteren Mausklicken wird das Bild langsam 'monitorgrün' und danach in verschiedenen Farbversionen dargestellt:

1) Rot 2) Blau 3) Gelb 4) Lila 5) Türkis

(Von jeder Farbe gibt es allerdings 2 verschiedene Arten.) Danach wird das Bild in mehreren Schritten in den ursprünglichen Zustand umgewandelt, worauf noch folgende Versionen der Darstellung folgen:

1) ABSTRACT I
2) ABSTRACT II
3) NEGATIV
4) ANTIQUE
5) MODERN
6) KOMPLEMENTÄR

und alles geht wieder von vorne los...

## Empfehlung:

- HAM-Bilder sollten natürlich nicht benutzt werden.
- Bei nichtfarbigen Bildern sind folgende Schritte effektlos:
a) Umwandlung zum SW-Bild (da es ja schon eins ist)
b) die 1. und 2.Art jeder Farb-Darstellung (grün,rot,blau, gelb, lila und türkis) wären identisch,
c) ABSTRACT I wäre die gleiche Darstellung wie ABSTRACT II, und
d) KOMPLEMENTÄR wäre die gleiche Darstellung wie NEGATIV.

- Am schönsten ist das Programm an (farbigen!) Bildern von Personen (Gesichtern) auszuprobieren

## Wie es funktioniert...

In der 1. Zeile werden die Variablen *ccr()*, *ccg()* und *ccr()* mit dem Index 31 dimensioniert. In ihnen sollen die Werte für die Rot-, Grün- und Blauanteile jeder Farbe 'abgelegt' werden. Dazu muß man diese Farbanteilwerte aber erst einmal wissen. Da es dafür keine Basic-Funktion (die eine entsprechend entgegengesetzte zu der "PALETTE"-Anweisung wäre) gibt, muß man zu "PEEK" greifen: In Zeile 2 wird der Variablen "cadr" die Adresse für die Werte der Farben zugewiesen. In den nächsten Zeilen werden dann diese Farbwerte in den Variablen *ccr()*, *ccg()* und *ccb()* gespeichert.

## Dann geht es los:

Um ein SW-Bild zu erhalten muß man den Durchschnitt der Variablen für die verschiedenen Farbanteile errechnen und diesen dann allen verschiedenen Farbanteilen wieder zuweisen, was mit der "PALETTE"-Anweisung erreicht wird. Um das Bild schrittweise umzuwandeln, wie es in diesem Programm der Fall ist, muß man erstmal den Durchschnitt der Farbanteile in einer Variablen (in diesem Fall "bw") ablegen und dann jeweils 'prozentual' einen gewissen Anteil des ursprünglichen Farbanteils und einen gewissen Anteil des Durchschnittswertes, also der Variablen ("bw") für die "PALETTE"-Anweisung benutzen, wobei der Anteil für die Variable ("bw") natürlich immer größer wird (und das Bild immer 'schwarzweisser'). Der Anteil wird im Programm durch die Variable "i" bestimmt; die Steigerung durch eine "FOR/NEXT" -Schleife. Bei Darstellung in 8- oder nur 4 Graustufen ist die Variable "bw" umso 'gröber', d.h. es wird nicht 16fach unterschieden, sondern nur 8 bzw. 4fach. Die 'Vergröberung' wird durch folgende Zeilen erreicht:

```
...
bw=bw*16
bw=INT(bw/i):bw=bw*i
bw=bw/16
...
```

Da "bw" ein Wert zwischen 0 und 1 ist, wird er mit 16 malgenommen, so daß man (Grau-) Werte zwischen 0 und 16 erhält.
Ist die Variable *i=1* (in den obenstehenden 3 Zeilen), wird im Prinzip überhaupt nichts verändert. "bw" wird zwar zu einer Ganzzahl, aber beschreibt immer noch einen Wert zwischen 0 und 16 (eigentlich müßte es ja heißen 0-15, da es nur 16 Grauwerte geben kann; die Werte 16 und 15 sind aber letztendlich gleich). Ist *i=2* , wird jeder "bw"-Wert durch 2 geteilt, aufgerundet und wieder mit 2 malgenommen. Dadurch erhält man nur gerade Werte wie z.B. 2, 4, 6, 12, 14 usw.; es gibt also nur noch 8 verschiedene Werte... Genauso verläuft es mit *i=4*. Bei den nun kommenden Darstellungen in Grün, Rot, Blau usw., wird folgendermaßen verfahren: Da es pro Farbe 2 Arten gibt, hier erstmal nur die 1.Art jeder Farbe:

Es wird wie bei der Schwarzweißdarstellung der Durchschnitt aller Farbanteilswerte errechnet. Die "PALETTE"-Anweisung lautet hier dann aber nicht:

```
PALETTE farbe,bw,bw,bw
```

was ein SW-Bild ergeben würde, sondern z.B.

```
PALETTE farbe,0,bw,0 'für Grün
PALETTE farbe,bw,0,0 'für Rot
...
```

d.h. verschiedene Farbanteile werden auf Null gesetzt, wenn sie nicht benötigt werden (bei Grün sind dies Blau+Rot; bei Türkis ist es nur Rot...)

Die 2.Art:

## Vorausüberlegung:

Um die ursprünglichen Farben wieder zu bekommen, könnte man einfach befehlen:

```
PALETTE farbe, ccr(farbe),
ccg(farbe), ccb(farbe)
```

Bei der 2.Art der Darstellung in verschiedenen Farben wird einfach folgendermaßen verfahren:

```
PALETTE farbe,0,ccg(farbe),0 'für Grün
PALETTE farbe,ccr(farbe),0,0 'für Rot
```

es wird also kein Durchschnitt benötigt, was zur Folge hat, daß das Bild nicht mehr so 'naturgetreu' ist (was es ja eigentlich sowieso nicht mehr ist), da alle anderen Farbwerte nicht mehr berücksichtigt werden. Außerdem wird es z.B. bei der Mischfarbe Lila dazu kommen, daß einzelne Farben blauer, andere roter sind, sie also nicht einheitlich lila in verschiedenen Helligkeitsstufen auf-

treten. Die anderen, wahrscheinlich noch effektvolleren, Darstellungsarten entstehen nur durch die gegebenen Werte für die Farbanteile und durch den daraus resultierenden Durchschnitt, also hauptsächlich durch die Variablen ccr(), *ccg()* und *ccb()*... Sie können mit dem Programm natürlich beliebig experimentieren. Viel Spaß bei der Bildmanipulation.

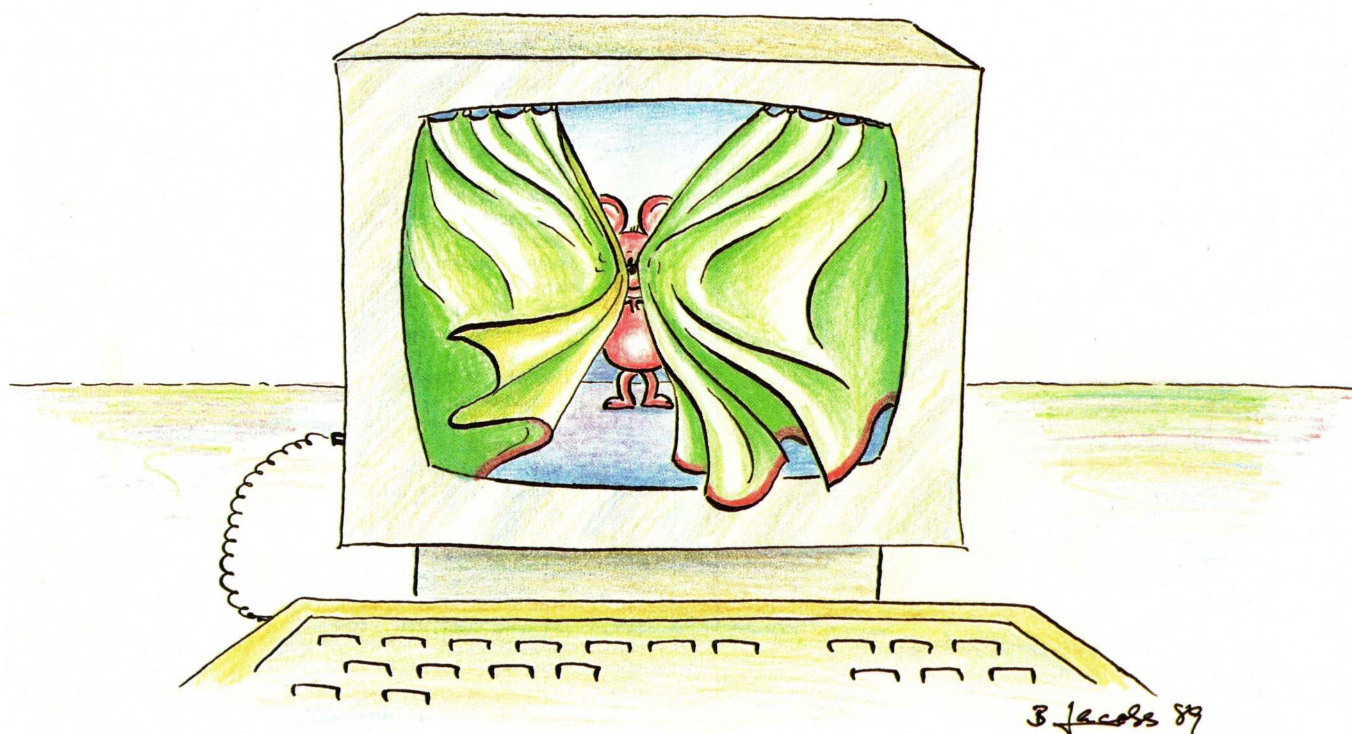
END of KICK

```
' Sprache: AMIGABASIC
' (c) MAXON Computer GmbH
' KICKSTART 1989
' Autor: Michael Teistler
DIM ccr(31),ccg(31),ccb(31)
cadr=PEEKL(PEEKL(PEEKL(WINDOW(7)+46)+48)+4)
FOR i=0 TO 31
 gb=PEEK(cadr+2*i+1)
 g=INT(gb/16):b=gb-g*16
 r=PEEK(cadr+2*i)
 ccr(i)=r/16
 ccg(i)=g/16
 ccb(i)=b/16
NEXT i
Gmain:
FOR i=100 TO 0 STEP -25
 FOR j=0 TO 31
  bw=ccr(j)+ccg(j)+ccb(j)
  bw=bw/3:ie=i/100:iz=1-ie
  r=ccr(j)*ie+bw*iz
  g=ccg(j)*ie+bw*iz
  b=ccb(j)*ie+bw*iz
  PALETTE j,r,g,b
 NEXT j
 WHILE MOUSE(0)>-1:WEND
NEXT i
i=1
FOR kk=1 TO 3
 FOR j=0 TO 31
  bw=ccr(j)+ccg(j)+ccb(j)
  bw=bw/3:bw=bw*16
  bw=INT(bw/i):bw=bw*i
  bw=bw/16:PALETTE j,bw,bw,bw
 NEXT j
 WHILE MOUSE(0)>-1:WEND
 i=i*2
NEXT kk
FOR i=100 TO 0 STEP -25
 FOR j=0 TO 31
  bw=(ccr(j)+ccg(j)+ccb(j))/3
  bwz=bw*16:bwz=INT(bwz/4)
  bwz=bwz/4
  f1=i/100:f2=1-f1
  r=f1*bwz:g=f1*bwz+f2*bw:b=f1*bwz
  PALETTE j,r,g,b
 NEXT j
 WHILE MOUSE(0)>-1:WEND
NEXT i
FOR j=0 TO 31
 PALETTE j,0,ccg(j),0
NEXT j
WHILE MOUSE(0)>-1:WEND
FOR j=0 TO 31
 bw=(ccr(j)+ccg(j)+ccb(j))/3
 PALETTE j,bw,0,0
NEXT j
WHILE MOUSE(0)>-1:WEND
FOR j=0 TO 31
 PALETTE j,ccr(j),0,0
NEXT j
WHILE MOUSE(0)>-1:WEND
FOR j=0 TO 31
 bw=(ccr(j)+ccg(j)+ccb(j))/3
 PALETTE j,0,0,bw
NEXT j
WHILE MOUSE(0)>-1:WEND
FOR j=0 TO 31
 PALETTE j,0,0,ccb(j)
NEXT j
WHILE MOUSE(0)>-1:WEND
FOR j=0 TO 31
 bw=(ccr(j)+ccg(j)+ccb(j))/3
 PALETTE j,bw,bw,0
NEXT j
WHILE MOUSE(0)>-1:WEND
FOR j=0 TO 31
 PALETTE j,ccr(j),ccg(j),0
NEXT j
WHILE MOUSE(0)>-1:WEND
FOR j=0 TO 31
 bw=(ccr(j)+ccg(j)+ccb(j))/3
 PALETTE j,bw*.8,0,bw*.8
NEXT j
WHILE MOUSE(0)>-1:WEND
FOR j=0 TO 31
 PALETTE j,ccr(j)*.8,0,ccb(j)*.8
NEXT j
WHILE MOUSE(0)>-1:WEND
FOR j=0 TO 31
 bw=(ccr(j)+ccg(j)+ccb(j))/3
 PALETTE j,0,bw,bw
NEXT j
WHILE MOUSE(0)>-1:WEND
FOR j=0 TO 31
 PALETTE j,0,ccr(j),ccb(j)
NEXT j
WHILE MOUSE(0)>-1:WEND
FOR i=100 TO 0 STEP -20
 f1=i/100:f2=1-f1
 FOR j=0 TO 31
  PALETTE j,f2*ccr(j),ccg(j),ccb(j)
 NEXT j
 WHILE MOUSE(0)>-1:WEND
NEXT i
FOR i=0 TO 31
 ab=ccr(i)+ccg(i)+ccb(i)
 ab=ab/3
 IF ab<.3 THEN
   PALETTE i,0,0,.7
 ELSEIF ab>=.3 AND ab<.6 THEN
  PALETTE i,1,0,0
 ELSEIF ab>=.6 THEN
  PALETTE i,1,1,0
 END IF
NEXT i
WHILE MOUSE(0)>-1:WEND
FOR i=0 TO 31
 r=ccr(i):g=ccg(i):b=ccb(i)
 IF r>g AND r>b THEN
  PALETTE i,1,0,0
 END IF
 IF g>r AND g>b THEN
  PALETTE i,0,1,0
 END IF
 IF b>r AND b>g THEN
  PALETTE i,0,0,1
 END IF
NEXT i
WHILE MOUSE(0)>-1:WEND
FOR i=0 TO 31
 bw=ccr(i)+ccg(i)+ccb(i)
 bw=bw/3
 PALETTE i,1-bw,1-bw,1-bw
NEXT i
WHILE MOUSE(0)>-1:WEND
FOR i=0 TO 31
 ob=ccr(i)+ccg(i)+ccb(i)
 ob=ob/3
 PALETTE i,ob*.7,ob*.6,ob*.4
NEXT i
WHILE MOUSE(0)>-1:WEND
FOR i=0 TO 31
 m=ccr(i)+ccg(i)+ccb(i):m=m/3
 PALETTE i,m,.5,m
NEXT i
WHILE MOUSE(0)>-1:WEND
FOR i=0 TO 31
 PALETTE i,1-ccr(i),1-ccg(i),1-ccb(i)
NEXT i
WHILE MOUSE(0)>-1:WEND
GOTO Gmain
```

VON STEFAN WINTERSTEIN

# FADESCREEN

## Wie im Fernsehn ...



*Neidvoll blickt Otto-Normal-User auf die Grafikzaubereien der Assembler-Gurus - welcher Hobbyprogrammierer träumt nicht davon, seinen Werken einen professionellen Touch zu verleihen? Zu den Feinheiten, die oft das "gewisse Etwas" eines Programms ausmachen, zählt unter anderem auch das weiche Ein- und Ausblenden des Bildschirms ("fading"), wie man das zum Beispiel von Spielen her kennt.*

Aus Geschwindigkeitsgründen kamen für solche Effekte bisher nur C und natürlich Maschinensprache in Frage. Doch mit GFA-BASIC bekommt der Programmierer erstmals ein Werkzeug "unter die Finger" (und in den AMIGA), das solche Finessen im "lahmen" Basic, und dann auch noch uncompiliert, möglich macht.

### Was kann FADESCREEN?

FADESCREEN ist eine GFA-BASIC-Unterroutine, die ohne Probleme in eigene Programme eingebaut werden

kann. Sie ist kurz, flexibel und nicht zuletzt ausreichend schnell, um einen wirklich weichen Überblendeffekt zu erzielen. Sie beherrscht Ausblenden ebenso wie Einblenden, beides in variabler Geschwindigkeit. Des weiteren ist es möglich, die Farben nach Schwarz oder nach Weiß hin überzublenden, so daß der Bildschirm sich entweder verdunkelt oder hell aufleuchtet. Als besonderes Bonbon unterstützt FADESCREEN den Extra-Half-Bright-Modus mit seinen 64 Farben.

## Benutzung von FADESCREEN

Hat man das Demoprogramm erfolgreich eingegeben (Kommentare können natürlich weggelassen werden), kann die FADESCREEN-Procedure als Block abgespeichert und im eigenen Programm über die MERGE-Funktion eingefügt werden.

Ein Aufruf von FADESCREEN sieht folgendermaßen aus:

fadescreenScreenNummer, einblenden!, hell!, Verzögerung, Farbpalette&())

### *Nun zu der Bedeutung der einzelnen Parameter:*

ScreenNummer - ist die Nummer des betroffenen Bildschirms; wichtig ist, daß dieser vor Aufruf von FADESCREEN mit OPENS geöffnet wurde. Anhand dieser Nummer findet FADESCREEN in den Betriebssystemstrukturen die Anzahl der dargestellten Farben.

*einblenden!* - ist kein Imperativ; "!" steht vielmehr für eine Boolean-Variable, die entweder den Wert TRUE (wahr) oder FALSE (falsch) annehmen kann. Setzt man sie auf TRUE, wird der Screen ein-, steht sie auf FALSE, ausgeblendet (eigentlich logisch, oder?).

*hell!* - ist ebenfalls vom Typ BOOL; bei FALSE wird der Screen dunkel, bei TRUE - na!? - richtig, hell (Wer hätte das gedacht!).

Verzögerung - dieser Wert bestimmt die Überblendgeschwindigkeit. Sinnvoll sind Werte zwischen 0 (maximale Geschwindigkeit) und 10, was stark von der Anzahl der dargestellten Farben abhängt.

```
'------------------------------------
'     DemoFadeScreen (GFA-Basic)
'
'     Autor: Stefan Winterstein
'     (c) MAXON Computer GmbH 1989
'------------------------------------
DIM farb&(64)                           ! benutzte Palette
'
bp&=4                                   ! Anzahl der BitPlanes (1-6)
OPENS 1,0,0,320,256,bp&,&H0             ! Screen öffnen,
GOSUB zeichnewas                        ! für EHB &h80 anhängen
PAUSE 100
hell!=FALSE                             ! Zuerst nach schwarz blenden
'
REPEAT                                  ! jetzt gehts los
  TITLEW #0,"Ausblenden..."
  PAUSE 50
  fadescreen(1,FALSE,hell!,0,farb&())!Fade out Screen 1, delay 0
  TITLEW #0,"Einblenden..."
  PAUSE 50
  fadescreen(1,TRUE,hell!,1,farb&())! Fade in  Screen 1, delay 1
  hell!=NOT (hell!)
UNTIL MOUSEK=1                          ! Linke Maustaste -> Abbruch
'
PROCEDURE zeichnewas
  OPENW 0                               ! großes Fenster öffnen
  TITLEW #0,"Zeichnen..."
  FOR i=0 TO 2^bp&                      ! alle Farben durchlaufen
    farb&(i)=i/2*256+255*INT(SIN(i/20))+i/2 ! Palette erstellen
    SETCOLOR i,farb&(i)                 ! in Array eintragen
    COLOR i                             ! aktuelle Farbe setzen
    PBOX 0,0,320-(i-1)*10*(32/2^bp&),256-(i-1)*7*(32/2^bp&)
  NEXT i                                ! nächste Farbe
  farb&(1)=&HFFF
  SETCOLOR 1,&HFFF                      ! weiße Schrift
  PRINT AT(11,15);2^bp&;"-farbiger Screen"! zur Kontrolle
RETURN
'
PROCEDURE fadescreen(snr,ein!,weiß!,delay%,VAR col&())
  ' snr    = Screen-Nummer
  ' ein!   = einblenden? (TRUE/FALSE)
  ' weiß!  = überblenden nach weiß? (TRUE/FALSE)
  ' delay% = Verzögerung (f. Speed)
  ' col&() = Farbpalette im Hauptprogramm (global)
  '
  ' Blendet Screen weich aus oder ein.
  ' Dieser muß zuvor mit OPENS geöffnet worden sein.
  ' Berechnet dazu für jeden Farbanteil jeder Farbe delta-Werte,
  ' (abhängig von Anzahl der frames), um die der jeweilige
  ' Farbanteil dann verändert wird.
  '
  DIM r|(31),g|(31),b|(31),dr(31),dg(31),db(31)! temporär
  LOCAL i&,f&,in,w&,depth,frames,anzcol,scr%  ! lokale Variablen
  '
  ' Initialisierungs-Teil:
  ' ----------------------
  frames=15                  ! In wievielen Schritten soll
  '                            das Überblenden ausgeführt werden?
  w&=-(weiß!*15)             ! weiß-Wert auf 0 oder 15 setzen
  in=(-1)^NOT (ein!)         ! Hilfsparameter: -1 oder 1
  scr%=SCREEN(snr)           ! scr = Pointer to ScreenStructure
  depth=PEEK(scr%+184+5)     ! Screen->BitMap->Depth gibt
  IF depth<6                 ! Anzahl der Bitplanes
    anzcol=(2^depth)-1       ! Anzahl der Farben feststellen
  ELSE IF (AND(CARD{scr%+44+32},&H80)<>0)!Screen->ViewPort->Modes
    anzcol=31                ! wenn &h0080 gesetzt -> EHB-Modus
  ELSE
    GOTO quit                           ! HAM -> Abbruch
  ENDIF
  FOR i&=0 TO anzcol                    ! alle Farben durchgehen
    r|(i&)=AND(col&(i&),&HF00)/256 ! rot-Anteil ausmaskieren
    g|(i&)=AND(col&(i&),&HF0)/16   ! grün     "
    b|(i&)=AND(col&(i&),&HF)       ! blau     "
    dr(i&)=(r|(i&)-w&)/frames*in   ! Schrittweite für rot rechnen
    dg(i&)=(g|(i&)-w&)/frames*in   !       "          grün    "
    db(i&)=(b|(i&)-w&)/frames*in   !       "          blau    "
    r|(i&)=r|(i&)*-(NOT (ein!))+w& ! Startwerte der Farbant.
    g|(i&)=g|(i&)*-(NOT (ein!))+w& ! 0,15 oder unverändert
    b|(i&)=b|(i&)*-(NOT (ein!))+w&
  NEXT i&                               ! nächste Farbe
  '  Hier beginnt das Überblenden:
  '  ------------------------------
  FOR f&=1 TO frames                    ! Überblend-Schritte
```

→

*Farbpalette&()* - muß ein Variablenfeld vom Typ WORD sein (Endung &), in dem die Farbpalette des Screens enthalten ist. Da FADESCREEN die aktuellen Bildschirmfarben verändert, dient dieses Array dazu, die wertvollen Originalfarben sicher aufzubewahren. ("Arbeiten Sie nur mit Sicherungskopien!")

*Ein Beispiel:*

```
fadescreen(2,FALSE,TRUE,1,farben&())
```

blendet Screen 2 mit einem Verzögerungswert von 1 nach Weiß aus, wobei die Farbpalette im Feld farben&() abgelegt ist.

## Tips

Wem die Geschwindigkeit bei 32 Farben und einem Delay-Wert von 0 noch nicht ausreicht, der kann in FADESCREEN die Konstante "frames" auf einen kleineren Wert als 15 setzen. Die Überblendung erfolgt dann in groberen Schritten, dafür aber entsprechend schneller.

Im Demoprogramm kann die globale Konstante *bp&* (gleich am Anfang) variiert werden; sie bestimmt die Tiefe des Demo-Screens und somit die Anzahl der Farben. Um im Demoprogramm den ExtraHalfbright(EHB)-Modus zu testen, muß *bp&* auf einen Wert von 6 gesetzt und dem OPENS-Befehl ein &h80 angehängt werden. Im EHB-Modus ist ein Überblenden nach Weiß nicht sinnvoll. Der Grund hierfür liegt in der speziellen Architektur dieser Darstellungsart: Trotz 64 Farben werden nur 32 Farbregister benötigt (mehr sind auch gar nicht vorhanden); die zusätzlichen Farben werden einfach durch Halbierung der Helligkeitswerte dieser 32 Grundfarben gewonnen (daher "Half-Bright" - halbe Helligkeit). Die Farben 32 bis 63 können also allenfalls ein mittleres Grau, aber nie Weiß erreichen.

Der HAM-Modus (4096) kann aufgrund seiner vertrackten Farborganisation nicht "gefaded" werden.

```
      FOR i&=0 TO anzcol                  ! Alle Farben durchgehen
        SETCOLOR i&,r|(i&)+f&*dr(i&),g|(i&)+f&*dg(i&),b|(i&+f&*db(i&)
        '                                   ! und verändern
      NEXT i&                               ! nächste Farbe
      ~Delay(delay%)                        ! Verzögerung
      ' DOS-Aufruf ist ca. 1 sek. schneller als PAUSE!
    NEXT f&                             ! Nächster Überblend-Schritt
    quit:
    ERASE r|(),g|(),b|(),dr(),dg(),db()     ! Arrays wieder löschen
  RETURN                                    ! und Tschüß!
```

## Programmtechnisches

Um universell einsetzbar zu sein, wurde FADESCREEN so flexibel wie möglich konzipiert. Gleichzeitig sollte es weitgehend unabhängig vom Hauptprogramm sein und dessen Daten nicht beeinflussen. Aus diesem Grund verwendet es nur lokale Variablen sowie einen Zeiger auf ein Variablenfeld *(VAR col&())*; Hilfsarrays werden vor Verlassen der Routine wieder gelöscht. Um festzustellen, um was für eine Art von Screen es sich handelt, befragt FADESCREEN die Intuition-Screenstruktur. 'C' bewältigt das zwar ungleich eleganter, doch auch GFA-BASIC bietet die Möglichkeit einer solch systemnahen Programmierung.

Apropos "systemnah": Wer sich wundert, warum ich nicht den PAUSE-Befehl als Verzögerung benutzte, dem sei gesagt, daß GFA-BASIC zur Ausführung von "PAUSE 0" schon zirka eine Sekunde benötigt. Deshalb entschied ich mich dafür, die "Delay"-Funktion der DOS-Library direkt anzuspringen *("~Delay(delay%)")*.

Doch nun zum verwendeten Algorithmus: Der Inhalt eines Farbregisters besteht aus einem 2-Byte-Wert, also 16 Bit, wovon jedoch nur die unteren zwölf genutzt werden. Davon bilden je 4 Bit den Rot-, Grün- und Blauanteil der Farbe. Herkömmliche Faderoutinen diverser Publikationen begnügen sich nun damit, jeden dieser drei Farbanteile solange um 1 zu erniedrigen, bis er bei 0 angelangt ist. Helle Farben brauchen also viel länger, um die Endfarbe (meist Schwarz) zu erreichen, als dunkle, was zu unschönen Farbverfälschungen führt und dem Betrachter nur ein gequältes Lächeln entlockt.

FADESCREEN geht hier einen etwas aufwendigeren, trotzdem lohnenden Weg: Für jeden Farbanteil einer jeden Farbe wird zunächst berechnet, wie weit er von der Zielfarbe der Überblendung entfernt ist. Ausgehend von diesem "Delta-Wert" wird errechnet, um welche Schrittweite der jeweilige Farbanteil bei jedem Darstellungsdurchlauf (frame) verändert wird. Jeder Farbanteil aller Farben erhält somit eine eigene "Geschwindigkeit", mit der er sich der Zielfarbe nähert, die dann von allen Farben gleichzeitig im letzten Frame erreicht wird. Alle diese Berechnungen werden mit Real-Zahlen ausgeführt, die der SETCOLOR-Befehl dann später selbsttätig auf Byte-Format zurechtstutzt. Erst wenn die Initialisierung der Hilfsarrays abgeschlossen ist (bei 32 Farben in ca. 0,4 Sekunden), beginnt die zeitkritische Überblendschleife, die aufgrund der zuvor berechneten Werte die neuen Farben in die Farbregister schaufelt.

Wer jetzt noch Fragen hat, sollte mal einen Blick ins Listing werfen, das ich hoffentlich ausreichend kommentiert habe. Vielleicht bringt es ja auch den einen oder anderen auf den Geschmack, sich mal wieder mit dem guten alten (GFA-)Basic zu beschäftigen.

END of KICK

VON HANS-H. ACKERMANN

# Variable Bildausschnitte

*Jedes anspruchsvolle Mal- und Grafikprogramm verfügt über eine Funktion, die es gestattet, Bildausschnitte von beliebiger Größe auf dem Bildschirm zu verschieben bzw. zu kopieren. Hierbei besteht der Unterschied zwischen "Verschieben" und "Kopieren" darin, daß bei der Funktion "Verschieben" der Ausschnitt im wahrsten Sinne aus dem Bild "herausgeschnitten" wird.*

Bei der Funktion "Kopieren" verbleibt der Originalausschnitt im Bild, die Kopie des Ausschnitts läßt sich an anderer Stelle beliebig positionieren.

Das Verschieben von grafischen Einheiten stellt den wesentlichen Bestandteil des sogenannten Layouts etwa einer Druckvorlage dar. Die in diesem Artikel gezeigten Routinen lassen Schere und Klebstoff - Hauptwerkzeuge beim Layout - endgültig überflüssig werden: Alle grafischen Einheiten, seien es Texte, Bilder oder auch Notensysteme, können mit der Maus auf dem Bildschirm druckfertig arrangiert werden.

## Das Prinzip

Der zu verschiebende Bildausschnitt wird mit dem "Gummifaden" eingerahmt. Dieser eingerahmte Bildschirmbereich wird in geeigneter Weise zwischengespeichert - gepuffert - und kann dann an beliebiger Stelle mit der Maus wieder auf dem Bildschirm plaziert werden. Bei der Funktion "Verschieben" wird der Bereich unter dem Rahmen gelöscht.

## Die Realisierung

Die Realisierung dieses an sich einfachen Prinzips findet sich in der *Procedure Verschieben(Copy%)*. Betrachten wir das Listing.

Die *Do-Loop*-Hauptschleife ist nur notwendig, falls nach dem Aufruf der Prozedur die Möglichkeit bestehen soll, mehrere Verschiebungen unmittelbar nacheinander vornehmen zu können.
Die erste *While-Wend*-Schleife wird bis zum Mausklick mit der linken Taste durchlaufen. Dabei ist die Menüüberwachung eingeschaltet. Erfolgt nun ein Mausklick, wird mit *MOUSE xl%, yl%,k%* die momentane Bildschirmposition der Maus festgehalten und der Variablen *xl%* bzw. *yl%* übergeben.

## Der Rahmen

Die Werte der Koordinaten *xl%* und *yl%* bleiben in der nun folgenden Rahmenschleife konstant, da hier mit *MOUSE xr%,yr%,k%* eine eigene Mausstatusauswertung erfolgt. Der Grund für diese zweifache Statusermittlung liegt in dem Prinzip, nach dem der "Gummifaden" auf dem Bildschirm erzeugt wird: Für das Zeichnen des Rahmens mit dem *BOX-Befehl* werden vier Koordinaten benötigt, zwei für die linke obere Ecke *(xl%,yl%)* und zwei für die rechte untere Ecke *(xr%,yr%)*. Da die Größe des Rahmens mit der Maus eingestellt wird und diese nur zwei Koordinaten liefert, müssen die Werte der linken, oberen Koordinaten konstant bleiben, während die Werte der Koordinaten der rechten, unteren Ecke analog zur Mausbewegung verändert werden können. Zur Realisierung des "Gummifadens" benötigt man demnach prinzipiell zwei Schleifen mit zwei unabhängigen Statusermittlungen.

Nach einer Bewegung der Maus muß vor dem Zeichnen des neuen Rahmens der alte Rahmen gelöscht werden, da andernfalls nach mehreren Mausbewegungen nur noch ein schwarz ausgefülltes Rechteck auf dem Bildschirm verbleiben würde. Das Löschen das alten Rahmens geschieht mit dem zweiten *BOX*-Befehl innerhalb der Rahmenschleife. Die beiden gleichartigen *BOX-Befehle* sind durch eine *RepeatUntil*-Schleife voneinander getrennt. Diese Schleife wird nach einer Mausbewegung verlassen, und das zweite *BOX xl%,yl%,xr%,yr%* überschreibt (im Graphmode 3) den alten Rahmen. Die Rahmenschleife wird verlassen, sobald die linke Maustaste losgelassen wurde.

Der eingerahmte Bereich wird jetzt mit dem Befehl *GET xl%,yl%,xr%,yr%, snp$* in die Stringvariable *snp$* kopiert. Bei zu großen Ausschnitten kann der String zu groß werden, was eine Fehlermeldung zur Folge hat. Man kann dies vermeiden, indem man, bevor der Ausschnitt gespeichert wird, die Größe des Ausschnitts berechnet und überprüft, ob es zu einem Stringüberlauf kommen kann. Die Berechnung sieht folgendermaßen aus:

```
DIV(MUL(MUL(SUB(xr%,xl%),SUB(yr%,yl%)),
    Screentiefe),8)<32676
  GET xl%,yl%,xr%,yr%,snp$
ELSE
  ERROR  'Fehlerroutine anspringen oder
         ALERT ausgeben
ENDIF
```

Soll der Ausschnitt verschoben, aber dabei nicht kopiert werden, überdeckt ein Rechteck von der Größe des Ausschnitts den eingerahmten Bereich mit *PBOX xl%,yl%, xr%,yr%* bei *Deffill 0*. Die Füllmusterfarbe wird danach sicherheitshalber mit *Deffill 1* wieder auf *"Objektfarbe"* festgelegt. Die Entscheidung, ob der Ausschnitt verschoben oder kopiert wird, übergibt man schon beim Aufruf der Prozedur mit dem Wert der Variablen *Copy%*.

## Verschieben und Kopieren

In der Verschiebeschleife wird nach erneuter Statusermittlung zunächst der Originalhintergrund mit *SGET Bild$( )* gepuffert. Nun zeigt *PUT x%,y%, Snp$,7* den Ausschnitt, was allerdings

```
RESERVE 200000
DIM m$(32),m1$(10)
OPENW #0
RESTORE mmenue
FOR z%=0 TO 10
  READ m$(z%)
NEXT z%
mmenue:
DATA GRAFIK  ,  Flächen  ,  Text , ---------------,  Quit,
DATA LAYOUT  ,  Verschieben     ,  Kopieren     ,,
MENU m$()
MENU KEY 1,70
MENU KEY 2,84
MENU KEY 7,86
MENU KEY 8,75
GOSUB menu
'
PROCEDURE menu
  DEFMOUSE 0
  DO
    SLEEP
    REPEAT
    UNTIL MOUSEK=0
    ON MENU GOSUB auswahl
  LOOP
RETURN
'
PROCEDURE auswahl
  MENU KILL
  SELECT MENU(0)
  CASE 1
    GOSUB flaechen
  CASE 2
    GOSUB text
  CASE 4
    END
  CASE 7
    GOSUB verschieben(1)
  CASE 8
    GOSUB verschieben(0)
  ENDSELECT
  MENU m$()
  MENU KEY 1,70
  MENU KEY 2,84
  MENU KEY 7,86
  MENU KEY 8,75
RETURN
'
'
PROCEDURE verschieben(copy%)
  DEFMOUSE 5
  WHILE MOUSEK=0 OR MOUSEK=2! solange keine oder rechte Maustaste
  WEND
  MOUSE xl%,yl%,k%
  GRAPHMODE 3
  WHILE MOUSEK=1
    MOUSE xr%,yr%,k%
    BOX xl%,yl%,xr%,yr%
    REPEAT
    UNTIL MOUSEK=0 OR MOUSEX<>xr% OR MOUSEY<>yr%
    BOX xl%,yl%,xr%,yr%
  WEND
  GET xl%,yl%,xr%,yr%,snp$
  IF copy%
    DEFFILL 0,1
    GRAPHMODE 0
    PBOX xl%,yl%,xr%,yr%
  ENDIF
  WHILE MOUSEK=0
    MOUSE x%,y%,k%
    SGET m1$()
    PUT x%,y%,snp$,224
    WHILE MOUSEX=x% AND MOUSEY=y% AND MOUSEK=0
    WEND
    SPUT m1$()
  WEND
  PUT x%,y%,snp$,224
  DEFMOUSE 0
RETURN

PROCEDURE flaechen
  GRAPHMODE 0
  COLOR 1
```

erst ab dem zweiten Schleifendurchlauf, d.h. nach der ersten Verschiebung, deutlich wird: Nach einer Mausbewegung wird die *While-Wend*-Warteschleife verlassen und mit *SPUT Bild$( )* der Ausschnitt an seiner alten Position überdeckt. Die Verschiebeschleife wird erneut durchlaufen, und jetzt zeigt der *PUT*-Befehl den Ausschnitt an seiner neuen Position.

```
  FOR z%=1 TO 5 STEP 2
    DEFFILL 1,2,RANDOM(24)
    PBOX 70*z%,20*z%,100*z%,60*z%
  NEXT z%
  DEFFILL 1
RETURN
'
PROCEDURE text
  GRAPHMODE 0
  COLOR 1
  FOR z%=20 TO 180 STEP 80
    TEXT 340,z%," Verschieben oder Kopieren"
  NEXT z%
RETURN
'
```

Erfolgt nun ein abschließender Mausklick, wird die Verschiebeschleife verlassen, und der Ausschnitt befindet sich an seiner endgültigen Position.

Die Möglichkeit zum Verschieben des Ausschnitts resultiert demnach aus dem Zusammenwirken der Befehle *SPUT* für den Hintergrund und *PUT* für den Ausschnitt. Die Tatsache, daß der Ausschnitt nach einer Mausbewegung an seiner alten Position mit *SPUT* überdeckt wird, ist übrigens die Ursache für ein leichtes Flackern des Ausschnitts während des Verschiebens. Die Warteschleife sorgt dafür, daß bei unbewegter Maus kein solches Flackern entsteht. Die beschriebene Prozedur kann in jedes Grafikprogramm integriert werden. In dieser Demoversion werden mit der *Procedure Flaechen* bzw. *Procedure Text* einfache grafische Einheiten zum Ausprobieren der Verschiebe- bzw. Kopierfunktion erzeugt. ■

# T&T

# Tips & Tricks

## MOUNT RAD

Ich habe noch einen kleinen Tip zur resetfesten RAM-Disk RAD: Man sollte die Befehle "*mount RAD:*" (anmelden) und "*dir RAD:*" (Sichtbarmachen auf der WB) in die Startup-sequence einbinden. Wenn man zuviel Speicher hat, kann man auch noch die RAD: in der Mountlist verändern (z. Bsp. mit "*ed devs/mountlist*"). Dort sieht man unter anderem bei *RAD LowCyl=0* und *HighCyl=21*. Bei *HighCyl* sollte man 21 in 79 ändern. Das Ganze abspeichern und rebooten. Nun sollte auf der WB ein Diskicon "*RAMBO*" (nicht der schielende Kinoheld, der noch dümmer ist, als er aussieht) erscheinen. Jetzt ist genug Platz auf der RAD:, um die ganze WB-Disk mit "*copy df0: to RAD: all*" auf RAD: zu kopieren. Anschließend die WB-Disk herausnehmen und rebooten. Jede selbstbootende Harddisk wird vor Neid erblassen ...

*(Andreas Herzog)*

## WBSTART

In den KICKS für Insider "Geheime Infos" (Heft 3/89, S.94) wird gezeigt, wie ein Programm in Modula feststellen kann, ob es von der Workbench oder vom CLI aus aufgerufen wurde. Dabei wurde die Funktion *GetLock* aus dem Modul "*Arguments*" verwendet. Dies ist erstens umständlich und erzeugt zweitens ein längeres Programm, da das gesamte Modul dazugelinkt wird. Es sind immerhin 1634 Bytes, die das Programm unnötig verlängern.

Es geht nämlich viel einfacher:
Bei M2Amiga von A+L stellt das Modul "*Arts*" die Variable "*wbStarted*" zur Verfügung. Folgendes Beispiel zeigt die Anwendung.
Das Modul "*Arts*" wird sowieso zum Programm gelinkt, da es das gesamte Laufzeitsystem enthält. Deshalb ist die hier vorgestellte Variante nicht nur einfacher, sondern auch kürzer als die im Heft 3/89 vorgestellte.

*(Thomas Stümpfig)*

```
...

FROM Arts IMPORT wbStarted:

...

IF wbStarted THEN WriteString ("Start erfolgte von Workbench aus.")
           ELSE WriteString ("Start erfolgte vom CLI aus.")
END;

...
```

## ARRAYS BEIM AZTEC C-COMPILER

Der ansonsten gute Aztec C-Compiler (ich rede von Version 3.6, aber für die vorhergehenden Versionen dürfte das gleiche gelten) hat einen Fehler, was die Behandlung sehr großer Arrays angeht. Da man diese nur selten braucht (z. B. beim "Sieb des Eratosthenes"), ist er wohl bisher nicht sonderlich aufgefallen. Aber man probiere einmal folgendes Mini-Programm:

```
char a[70000L];
main ()
{
register long i;
  for (i=0;i<70000L;a[i++]=2;
}
```

Ein Absturz ist beim Ausführen ziemlich sicher. Jawohl Absturz, der Guru kommt nämlich gar nicht mehr zu Wort. Um dem Fehler auf die Spur zu kommen, compiliere man nun mit "*-a-t*" und schaue sich den entstandenen Assemblerquelltext an. Und was sieht man (staun!):

```
global a,44464
```

Das Array *a* wird statt auf 70000 nur auf 4464 (=70000-65536=70000-216) Bytes Größe dimensioniert, so daß das Programm sehr schnell in irgendwelchem Speicher rumschreibt.
Einwandfrei haben hier die Entwickler des Compilers ein wenig Platz und Zeit sparen wollen und der Routine, die für die Größe von Arrays zuständig ist, nur eine short-Variable für die Größe spendiert. So ärgerlich dieser Fehler ist, so leicht läßt er sich umgehen. Das unten stehende Programm läuft hervorragend (wenn es auch - genauso wie obiges - keinen tieferen Sinn hat).

```
extern void *AllocMem();
char *a;

main ()
{
register long i;
  a=AllocMem (70000L,0L);
  for (i=0;i<70000L;a[i++]=2);
  FreeMem (a,70000L);
}
```

Natürlich sollte man bei eigenen Programmen die Sicherheitsabfrage bei *AllocMem* nicht vergessen.

*(Markus Baumeister)*

# Probleme beim Grafikdruck

Beim Ausdrucken von Grafikbildern steht man oft vor dem Problem, den Druck vorzeitig abbrechen zu müssen, weil der Drucker noch nicht eingestellt, oder das Resutltat nicht so ist, wie gewünscht. Eine Möglichkeit ist, den Drucker Offline zu schalten, aber dann tritt eine Wartezeit von 30 Sekunden ein, bis der Requester erscheint.

Abhilfe schafft da ein Patch für den Druckertreiber: Bei fast allen Treibern ist das Byte 149, das dez 30 enthält, für 30 Sekunden Timeout verantwortlich.

Mit dem beiliegenden kleinen C-Programm kann das Byte z.B. in 10 (10 Sekunden) geändert werden. Das Programm eignet sich natürlich auch für andere Patchfälle.

*(Wolfgang Schilling)*

```
#include <exec/exec.h>
#include <stdio.h>

#define BUFFER 102400
/*  Genau 100k puffer wird reserviert */

int *speicher, zaehler=0;
char tname[80];

main()
{
speicher = AllocMem(BUFFER,MEMF_PUBLIC);
if(speicher==NULL)
    {
      puts("Nicht genug Speicher");
      CloseAll();
    }
printf("\nBitte Pfad- und Dateiname eingeben:  ");
scanf("%s",&tname);

Laden();
Patch();
Speichern();
CloseAll();
}

CloseAll()
{
   FreeMem(speicher,BUFFER);
   exit(0);
}

Laden()
{
FILE *datei;
if ((datei = fopen(tname,"r")) == NULL)
   {
    puts("Nicht lesbar!");
    CloseAll();
   }
while((*(speicher + zaehler) = fgetc(datei))!= EOF)
         zaehler++;
  fclose(datei);
  printf(2\nDie Datei ist %d Bytes lang! \n",zaehler);
  }

  Speichern()
  {
  FILE *datei;
  int error = 0, count = 0;
  strcat(tname,".PAT");
  if ((datei = fopen(tname,"w")) == NULL)
     {
       puts("Nicht beschreibbar!");
       CloseAll();
     }
  while(count < zaehler)
     {
       error = fputc(*(speicher + count),datei);
       count++;
     }
  fclose(datei);
  printf("Save %d Bytes.\n\n",zaehler);
  }

  Patch()
  {
  int address, patches;

  for(;;)
   {
      printf("Bitte Adresse eingeben, wo geaendert
                  werden soll (ENDE mit -1):\n");
      scanf("%d",&address);
      if(address == -1)
        break;
      printf("Alter Wert ist dezimal %d\n",*(speicher
                                   + address));
      printf("Neuer dezimaler Wert eingeben: \n");
      scanf("%d",&patches);
      *(speicher + address) = patches;
   }
  }
```

# Externes Laufwerk an XT-Karte?!

Wenn Sie auch stolzer Besitzer einer XT-Karte sind, haben Sie auch vielleicht schon ohne Erfolg versucht, Ihr externes 3,5"-Laufwerk an der XT-Karte zu betreiben. Timing-Probleme haben möglicherweise ein Laufen bisher verhindert. Wenn Sie die folgende Bastelanleitung benutzen, bestehen gute Aussichten, daß auch Ihr externes Laufwerk in Zukunft einwandfrei arbeitet.

Um das Signal, das an PIN 21 (Drive Select 2) gelangt, zu verzögern, benötigen Sie lediglich einen kleinen Kondensator (470pF, 10Volt oder größer), den Sie innerhalb des Laufwerksgehäuses zum einen an PIN 21, zum anderen entweder an PIN 3, 4, 5, 6 oder 7 (alles Masse bzw. Ground) anschließen. Um herauszufinden, welches die richtigen Anschlüsse sind, können Sie den Laufwerksstecker öffnen und anhand der unterschiedlichen Farben der Drähte verfolgen, wo im Laufwerk auf der Interface-Platine die beiden entsprechenden Kabel ankommen. Falls Sie über ausreichende Geschicklichkeit verfügen und der Kondensator klein genug ist, können Sie ihn auch im Laufwerksstecker direkt an PIN 21 und Masse (PIN 3 - 7) anlöten, wobei Sie unbedingt Kurzschlüsse vermeiden sollten.

Bedenken Sie, daß durch das Öffnen des Laufwerksgehäuses die Garantie verlorengehen kann!

*(Werner Zempelin)*

# GFA BASIC

## Informationen die nicht im Handbuch stehen

Der GFA BASIC-Interpreter läßt einem das langsame AmigaBasic vergessen. Das Handbuch erklärt aber die Fülle an Befehlen und Funktionen nur dürftig. Hier deshalb einige Tips und Tricks zu GFA BASIC.

Beim Laden und Speichern von Programmen erscheint ein Dateiauswahlrequester, welcher Sie auch mit dem Befehl FILESELECT programmieren können. Directories erscheinen dort mit einem Sternchen vor dem Namen. Was aber, wenn man aus Versehen ein Directory angewählt hat, das man gar nicht wollte? Wenn Sie das Kästchen mit dem Kreuz in der oberen linken Ecke anklikken, gelangen Sie wieder in das übergeordnete Directory.

Wenn man mehrere Fenster eröffnet, kann man mit den Befehlen FRONTW und BACKW ein Fenster nach vorne oder hinten verlagern. Damit ein Fenster aktiv wird, ohne mit der Maus angeklickt werden zu müssen, benutzen Sie die Betriebssystemroutine *ActivateWindow*. Die Syntax lautet:

```
~ActivateWindow(WINDOW(nr)).
```

Mit *nr* geben Sie die Nummer des zu aktivierenden Fensters an.

Die Betriebssystemroutinen lassen sich mit GFA BASIC besonders leicht benutzen. Es braucht nicht wie beim AmigaBasic mit dem Befehl LIBRARY eine Bibliothek eröffnet zu werden. Es muß lediglich das Zeichen '~' vor der Routine stehen. Das aber nur, wenn die Routine keinen Wert zurückgibt, z.B.:

```
wert%=SystemRoutine(parameter)
```

erfolgt ohne das Zeichen '~'.

Bleiben wir noch bei den Fenstern. Wenn Sie die Zeichenausgabe auf ein anderes Fenster verlegen möchten, bedienen Sie sich des Befehls RASTPORT, z.B. RASTPORT LPEEK(WINDOW(nr)+50). Nun funktionieren die Zeichenbefehle in dem mit *nr* angegebenen Fenster. Wenn Sie aber einen eigenen Screen benutzen und den Rastportzeiger verändern, wird nach dem Schließen der Fenster und des Screens der Rastportzeiger nicht automatisch wieder angepaßt. Sie müssen ihn wieder auf den aktuellen Screen-Rastport setzen, um keinen Systemabsturz zu erleben. Das können Sie folgendermaßen erreichen:

```
AktuellerScreen%=LPEEK(_INTBase)+54
RASTPORT AktuellerScreen%+80
```

Mit den Befehlen MOUSEX und MOUSEY können Sie Mauskoordinaten abfragen. Wenn Sie mehrere Fenster geöffnet haben, beziehen sich diese Werte immer auf das zuletzt geöffnete Fenster, egal ob es aktiv ist oder nicht. Die Mauskoordinaten für ein beliebiges Fenster ermitteln Sie wie folgt:

```
mausx=DPEEK(WINDOW(nr)+14)
mausy=DPEEK(WINDOW(nr)+12)
```

In *nr* setzen Sie die Nummer des gewünschten Fensters.

Apropos Maus: Manchmal wäre es ganz schön, wenn man den Mauszeiger verschwinden lassen könnte, z.B. um in einem Grafikprogramm ein eigenes Fadenkreuz zu benutzen. Mit unten stehendem Unterprogramm können Sie die Maus verschwinden lassen. In *nr* setzen Sie die Nummer des Fensters, in dem der Mauszeiger verschwinden soll.

Die Version 3.02 von GFA BASIC erstellt jetzt auf Wunsch automatisch ein Icon. Damit läßt sich ein Programm von der Workbench aus starten. Dies funktioniert aber nicht, wenn das BASIC-Programm in einem Directory liegt. Dies können Sie ändern, wenn Sie das Icon anklicken und den Menüpunkt Info aus dem Workbench-Menü anwählen. Setzen Sie nun vor GFABASIC einen Doppelpunkt. Sie könnten auch GFA-BASIC in GFABASRO ändern, damit dieses Programm mit dem Run Only-Interpreter lauffähig ist.

Das war's! Ich hoffe, daß Sie etwas von diesen Tips gebrauchen können und wünsche viel Spaß bei der Programmerstellung mit GFA BASIC.

*(Herbert Kunz)*

```
PROCEDURE Mausweg(nr)
    Maussp%=MALLOC(12,&H10002)
    ~SetPointer(WINDOW(nr),Maussp%,1,16,250,250)
RETURN
```

Zum Einschalten dient folgende Routine:

```
PROCEDURE Mausan(nr)
    ~ClearPointer(WINDOW(nr))
    f%=MFREE(Maussp%,12)
RETURN
```

# Tips zu Ultima III

Beim Lesen der ULTIMA III-Lösung in der Kickstart 4/89 sind mir noch weitere hilfreiche Tips eingefallen:

1. Man sollte sich nur mit einem genügend großen Vorrat an KEYS und GELD in den Strudel, sprich: nach Ambrosia, begeben, da man dieses Land ohne Schlüssel (bzw. UNLOCK-Spell) nicht mehr verlassen kann; außerdem geht die Erhöhung von Persönlichkeitsattributen ganz schön ans Geld (100g pro Punkt).

2. Für Spieler niedrigerer Levels - und ohne moralische Bedenken - lohnt sich vielleicht der Hinweis, daß es in der Stadt YEW (Stadt der Clericer) keine GUARDS gibt. Und man sagt, daß die Clericer einiges an Geld und Waffen mit sich rumschleppen - aber, wie gesagt, das muß jeder mit seinem eigenen Gewissen vereinbaren. Die Stadt liegt in einem kleinen dunklen Wald, der nur von Westen her zu betreten ist.

3. Apropos Moral: Wer genug KEYS, ein Schiff, aber wenig Geld (minimum 100) hat, sollte der Stadt DEATH GULCH mal einen (oder auch mehrere) Besuch(e) abstatten. Sie ist nur mit Schiff zu erreichen, liegt aber dennoch ironischerweise nur ein Seefeld vom Festland entfernt auf einer Insel. In dieser Stadt gibt es im Süden, hinter einem Verkaufsstand, eine Schatzkammer, die einige CHESTS enthält. Der Eingang liegt im südöstlichen Teil der Stadt in einem Labyrinth. Der Zugang wird von einem Guard bewacht. Dieser verzieht sich aber wenn man ihn besticht (Other Cmd: BRIBE). Man sollte ihn jedoch um keinen Preis angreifen, denn in diesem Labyrinth hat man bei einem Kampf schlechte Karten, es sei denn man ist stark genug, aber auch dann kostet so ein Kampf nur unnötig Zeit.
Man kann beim Betreten der Stadt einen Schlüssel sparen, indem man nach dem Hineingehen (noch vor dem Dämon) sofort nach Süden geht. Dort befindet sich der Eingang zu einem kleinen Labyrinth, welches die Eingangstür umgeht. Den Weg findet man entweder durch Ausprobieren oder per GEM.

*(Michael Bernhardt)*

# Wiederbelebung bei Ultima-IV

Wenn man Lord British auf *"Health"* anspricht und dann mit *"NO"* antwortet, wiederbelebt und heilt er alle Charaktere.

# Elite

Man gebe bei der 1. Abfrage des Textes aus der Anleitung das Wort *"SARAH"* ein. Der Computer quittiert das als Fehler. Bei der zweiten Passwortabfrage antwortet man korrekt. Wenn man nun im Inventory die Taste *"HELP"* drückt, gelangt man in eine Art Monitor, mit welchem man gezielt wichtige Raumschiffdaten und Eigenschaften ändern kann.
Welche Bytes wofür zuständig sind, möchte ich hier nicht sagen. Viel Spaß beim Ausprobieren. Aber Vorsicht! Nicht zuviel auf einmal ändern. Bei Eingabe von undefinierten Werten nimmt das Programm seinen Hut.

*(Michael Bernhardt)*

# Codes zu "Bombuzal"

Ich habe alle Codes zu "Bombuzal". Sicherlich werden Sie dem einen oder anderen eine wertvolle Hilfe sein, besonders wenn man einmal ein Level satt hat und einfach im nächsten weiterspielen möchte.

| CODE | STARTLEVEL | CODE | STARTLEVEL | CODE | STARTLEVEL |
|---|---|---|---|---|---|
| BOMB | 1 | OPAL | 88 | SPOT | 176 |
| ROSS | 8 | SONG | 96 | PALM | 184 |
| RATT | 16 | FIRE | 104 | LOCK | 192 |
| LISA | 24 | LAMP | 112 | SAFE | 200 |
| DAVE | 32 | TREE | 120 | WORM | 208 |
| IRON | 40 | SINK | 128 | NOSE | 216 |
| LEAD | 48 | BIKE | 136 | EYES | 224 |
| WEED | 56 | BIRD | 144 | HAIR | 232 |
| RING | 64 | TAPE | 152 | SIGN | 240 |
| GIRL | 72 | VASE | 160 | MYTH | 248 |
| GOLD | 80 | PILL | 168 | | |

*(Andreas Herzog)*

# Tips & Tricks

Diese Rubrik dient dazu, wichtige Informationen an den Leser zu bringen. Daß dafür die Mitarbeit aus der Leserschaft erforderlich ist, versteht sich von selbst.
Falls Sie (Ja, genau Sie, nicht die Person hinter Ihnen!) noch etwas in der Trickkiste haben, so holen Sie es schnellstens heraus und senden es uns zu. Einsendungen werden natürlich honoriert.

*MAXON Computer*
*Redaktion KICKSTART*
*'Tips&Tricks'*
*Industriestr. 26*
*6236 Eschborn*

# WBMono

Mit dem folgenden Programm können rund 20 kByte Speicher eingespart werden, wenn man sich mit einer zweifarbigen Workbench (weiß auf blau) begnügt. Die Workbench setzt sich aus zwei Bitplanes zusammen, von denen jede ca. 20 kByte Speicher benötigt. Für jeden Bildpunkt lassen sich mit zwei Bits vier Farben darstellen. Das Programm "WBMono" schaltet beim ersten Aufruf eine Bitplane ab und stellt ihren Speicher dem System zur Verfügung. Der freie Systemspeicher wächst um 20 kByte, und der neue Workbenchscreen ist nun zweifarbig, da jetzt für jeden Bildpunkt nur noch ein Bit zur Verfügung steht. Zur Arbeit mit anderen Programmen steht jetzt mehr Speicher zur Verfügung. Außerdem werden die Textausgabe und das Scrolling in CLI-Fenstern beschleunigt. Ruft man das Programm ein zweites Mal auf, wird der alte Zustand wieder hergestellt (vierfarbige Workbench, 20 kByte weniger Speicher). Da der neue Bitplanespeicher vor seiner Zuweisung gelöscht worden ist, sind die Icons immer noch monochrom; nur der Füllbalken einer Diskette weist auf eine vierfarbige Workbench hin. Wird ein Icon angeklickt oder ein neues Fenster geöffnet, ist es neu "eingefärbt". Das Programm wurde in der Programmiersprache "Aztec C" erstellt. Mit wenigen Änderungen müßte es auch mit Lattice C zu compilieren sein.

*(Michael Maier)*

```
/*******************************/
/*      Monobitmap V1.0        */
/*-----------------------------*/
/* (C) 1989 by Michael Maier */
/*******************************/


#include <functions.h>
#include <exec/types.h>
#include <exec/memory.h>
#include <intuition/intuition.h>
#include <graphics/gfxbase.h>

#define PLANEMEMORY 20480L

struct GfxBase *GfxBase;
struct IntuitionBase *IntuitionBase;

main()
{
 if ((GfxBase=(struct GfxBase*)
OpenLibrary("graphics.library",0L))==NULL)
    exit(FALSE);
 if ((IntuitionBase=(struct IntuitionBase*)
    OpenLibrary("intuition.library",0L))==NULL)
    {
       CloseLibrary(GfxBase);
       exit(FALSE);
    }

 if (GfxBase->ActiView->ViewPort->RasInfo->BitMap->Depth!=1L)
    {
       GfxBase->ActiView->ViewPort->RasInfo->BitMap->Depth=1L;
       FreeRaster(GfxBase->ActiView->ViewPort->RasInfo->BitMap
       ->Planes[1],640L,256L);
       RemakeDisplay();
    }
 else
    {
       GfxBase->ActiView->ViewPort->RasInfo->BitMap->Depth=2;
       if((GfxBase->ActiView->ViewPort->RasInfo->BitMap->Planes[1]
         =AllocMem(PLANEMEMORY,MEMF_PUBLIC | MEMF_CHIP | MEMF_CLEAR))
                                                          ==NULL)
           {
              GfxBase->ActiView->ViewPort->RasInfo->BitMap->Depth=1;
              exit(FALSE);
           }
       RemakeDisplay();
    }
 CloseLibrary(GfxBase);
 CloseLibrary(IntuitionBase);
}
```

# Einkaufsführer
## Hier finden Sie Ihren Commodore/Amiga Fachhändler

### 1000 Berlin

HD AMIGA Hardware Spezialist
**Computertechnik**
Sämtliche verfügbaren KICKSTART-PD's vorrätig!
Und über 1500 weitere PD's für den AMIGA!
1000 Berlin 65 · Pankstr. 42
Tel. 030/465 70 28-29

COMPUTER-STUDIO
**Schlichting**
... die etwas andere Computerei
ATARI-Fachmarkt
MS-DOS Fachmarkt · NEC-Fachhandel
Katzbachstraße 8 · 1000 Berlin 61
☎ 030/7864340

Bei uns werben bringt

GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057

**SOFTPOWER**
Das Software-Paradies im Norden Berlins!
Mo-Fr. 11-18.30
Sa. 10-14
Daily News !!!
IBM
1000 Berlin 65
Schwedenstr. 18c
Tel.: 030/492 20 56

### 2000 Hamburg

**Bit Computer Shop**
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/494400

**Createam**
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon: 040/6416473+6416861

**GMA** mbH
040/251241 6717
Systemhändler
Wandsbeker Chaussee 58
2000 Hamburg 76

### 2000 Hamburg

**Computer**
Hardware · Software · Zubehör
Lilienstraße 32
(beim Mönckebergbrunnen)
2000 Hamburg 1
Tel. (040) 336708
SYSTEMSHOP ®

### 2160 Stade

**BERGAU**
Büromaschinen · EDV-Systeme
Neue Straße 5 · 2160 Stade
Telefon (04141) 2364 u. 2384

### 2300 Kiel

Hardware
Software
Service
**Home Computer Laden**
Spezialisiert auf Public Domain
Immer die neueste Software auf Lager
Gutenbergstraße 5 · 2300 Kiel · Tel. (0431) 555555

### 2390 Flensburg

ecl electronic computer laden ohg
Norderstraße 94-96 · D-2390 Flensburg
☎ (0461) 28181 + 28193

### 2900 Oldenburg

**GOLDT**
**Computerhaus**
Donnerschweer Straße 127-129
(gegenüber Weser Ems Halle)
2900 Oldenburg
Telefon (0441) 884706

### 2940 Wilhelmshaven

**Radio Tiemann**
Commodore-Systemfachhändler
Marktstraße 52
2940 Wilhelmshaven
Telefon (04421) 26145

### 3000 Hannover

**COM DATA**
Am Schiffgraben 19 · 3000 Hannover 1
Telefon (0511) 326736

Fun Action **CSJ** Fun Action
Hannover's Softwarethek Nr. 1
DIE AUSWAHL an Software für C 16, C 64 / 128, Amiga, Atari ST u. IBM PC.
An der Tiefenriede 27 · 3000 Hannover 1
Tel. 0511/88 63 83

### 3500 Kassel

**Hermann Fischer GmbH**
Commodore-Systemfachhändler
Rudolf-Schwander-Str. 5-13
3500 Kassel
Telefon (0561) 700000

### 4500 Osnabrück

**B.O.S.S.**
**Computerhaus**
Am Berliner Platz
Goethering 3
4500 Osnabrück
Telefon (0541) 26570

Bei uns werben bringt

GEWINN

Sprechen Sie mit uns.
Heim Verlag ☎ 06151/56057
BUF

## 4650 Gelsenkirchen-Horst

**MENTIS GmbH**

Hard- und Software, Literatur
Bauteile, Service, Versand
**Groß- und Einzelhandel**

Poststraße 15 · 4650 Gelsenkirchen-Horst
Telefon (02 09) 5 25 72

## 6200 Wiesbaden

Poststraße 25
Luisenstraße 47
6200 Wiesbaden
Telefon (0 61 21) 56 20 91




Werbung & EDV GmbH

AUTORISIERTER
COMMODORE
SYSTEM-HÄNDLER



Commodore

***COMPUTER TREFF***

*Computerbedarf, PD und*
*Software für*
*ATARI, AMIGA, PC*

*Nettelbeckstraße 12*
*6200 Wiesbaden*
*Tel. (0 61 21) 40 43 02*

## 6380 Bad-Homburg

**PDC GmbH**
**Produkte u. Details Computerverband**

Louisenstraße 115
Ladenpassage Alter Bahnhof
6380 Bad-Homburg
Telefon (0 61 72) 2 47 48



Autorisierter Commodore-Systemhändler!

## 6457 Maintal

**Landolt-Computer**

Beratung · Service · Verkauf · Leasing



Autorisierter Commodore-Händler

Wingertstr. 112 · 6457 Maintal/Dörnigheim
Telefon (0 61 81) 4 52 93

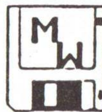
**Bei uns werben bringt**

**GEWINN**

Sprechen Sie mit uns.
Heim Verlag ☎ 0 61 51 / 5 60 57

BUF

## 6551 Fürfeld




MICHAEL WEISGERBER

HARD + SOFT

Rathausstraße 2
6551 Fürfeld
Telefon (0 67 09) 7 78

## 6680 Neunkirchen

**SHOP 64**
**Computer GmbH**
Neunkirchen
Telefon (0 68 21) 2 37 13

**Commodore Systemhändler**

Saarbrücken
Saarlouis
Homburg
St. Ingbert
Neustadt
Trier
Kaiserslautern

## 6800 Mannheim

GAUCH + STURM

Computersysteme + Textsysteme

**6800 Mannheim 24**
Casterfeldstraße 74-76
Telefon (06 21) 85 00 40 · Teletex 6 211 912

## 7000 Stuttgart

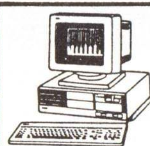
**»If AMIGA, go to Schreiber«**
Stuttgart's starker Computer-Laden.

**SCHREIBER**
COMPUTER

**Rotebühlplatz 10**
**7000 Stuttgart-1**
**Tel. 0711/227099**

**Neu · Neu · Neu · Neu**
**Im SUBWAY**
**BREUNINGER City**

## 7800 Freiburg



**Comp.Z.**

*Pochgasse 31*
*7800 Freiburg*
*T. 0761/554280*



Commodore

AUTORISIERTER
COMMODORE
SYSTEM-HÄNDLER

## 7890 Waldshut-Tiengen

**hettler-data**
**service gmbh**

Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon (0 77 51) 30 94

## 8000 München

**Commodore**

**AMIGA + PC, HARD- UND SOFTWARE**
***— BERATUNG / VORFÜHRUNG —***
***VIDEOBEARBEITUNG***

**MODL**
FOTO · VIDEO · AUDIO · FERNSEH
COMPUTER · GELEGENHEITSMARKT
REPARATUREN · FOTOKOPIEN
FARBKOPIEN · PASSBILDER
MAX-WEBER-PLATZ · 8000 MÜNCHEN 80 · ☎ 4 80 16 50

## 8400 Regensburg

**Zimmermann**
**elektroland**

**8400 Regensburg**
**Dr.-Gessler-Str. 8**
**☎ 09 41 / 9 50 85**

**8390 Passau**
**Kohlbruck 2a**
**☎ 08 51 / 5 20 07**

## 8700 Würzburg

**SCHOLL**
**BÜROTEAM**

Hardware · Software
Service · Schulung

**computer center**

am Dominikanerplatz
Ruf (09 31) 3 08 08 - 0

## Schweiz

Computer Trend

**Ihr Computer Spezialist**

**5000 Aarau,** Bahnhofstrasse 86,
Tel. 064/22 78 40
**4102 Basel-Binningen,** Kronenplatz,
Tel. 061/47 88 64
**5430 Wettingen,** Zentralstrasse 93,
Tel. 056/27 16 60
**8400 Winterthur,** St. Gallerstrasse 41,
Tel. 052/27 96 96
**8021 Zürich,** Langstrasse 31,
Tel. 01/241 73 73

**Grösste Auswahl an Peripherie, Software, Literatur und Zubehör.**

**ATARI**

**Commodore**

»Flickerfixer«
Endlich Schluß mit dem Interfaceflimmern!

»Flickerfixer«
Die Videokarte für A 2000 zum Anschluß an Multisyncmonitor

»Flickerfixer« ist für 1289,- zu haben.

Weiterhin im Programm:

| | |
|---|---|
| Kickstartumschaltplatine mit Kick 1.3 EPROMs | 160,- |
| Expansionsboard = zusätzliche Steckplätze für A 2000 PC-Teil | 165,- |

Exklusiv bei:

Der Computerladen
H & P Rodat
**CompiMate**

Versand per UPS
Nachnahme 10,-

Sudbrackstraße 31 • 4800 Bielefeld • Telefon 0521/133621

---

| | | |
|---|---|---|
| 2 MB-Rambox A1000 mit Bus 2 MB bestückt | DM | 949,00 |
| Amstrad LQ 3500 Letter-Quality-24-Nad-Drucker | DM | 599,00 |
| Aztec C Dev. V3.6 mit Source Level Debugger | DM | 550,00 |
| Aztec C Professional System V3.6 | DM | 279,00 |
| Digi View Gold V3.0 für A500/2000 | DM | 285,00 |
| Golem 2 MB-Rambox A1000/A500 | DM | 989,00 |
| Golem 20 MB-Harddisk A500/A1000 | DM | 899,00 |
| Golem 3.5-Zoll-Laufwerk | DM | 249,00 |
| Vortex 20 MB-Festplatte | DM | 935,00 |
| MCC-Makroassembler | DM | 79,00 |
| Cambridge Lisp/Metacomco | DM | 199,00 |
| TDI-Modula Developers Version V3.01 | DM | 199,00 |
| Silent Service/Thexder/Barabarian (Psyg.) | á DM | 49,95 |
| Guild of Thieves, Obliterator, Balance of P. | á DM | 49,95 |

Kostenlose Prospekte
auch für ST und IBM von

CWTG

**C W T G Joachim Tiede**
**Bergstraße 13 · 7109 Roigheim**
**Tel/BTX 06298/3098 von 17–19 Uhr**

---

**Achtung ★ Achtung ★ Achtung**

# Amiga-Programmierer

Wir suchen Amiga-Programme aus allen Bereichen (Grafik, Anwendung, Spiele usw.).
Assembler- und C-Programme bevorzugt!

**Wir bieten gute Umsatzbeteiligung!**

Sollten Sie Interesse haben, wenden Sie sich noch heute an:




**Software 2000** A. Wardenga
Lübecker Straße 10, 2320 Plön
Tel. 04522/1379
Fax: 04522/3296

---

**2000 Hamburg**

**BRINKMANN'S Computer Ranch**

SPEZIALISTEN für
AMIGA
und
PC – SYSTEME

Commodore

Mühlendamm 2
2000 HAMBURG 76
Tel.: 040/252557

---

# VESALIA-VERSAND

## Vesalia Harddisk für Amiga 500/1000

| | | |
|---|---|---|
| 21 MB | Schneller Treiber mit Autoinstallation, | 978,– |
| 31 MB | keine Programmierkenntnisse erforderlich. | 1098,– |
| 41 MB | Software verwaltet 2 Harddisk. Erkennt | 1478,– |
| 62 MB | u. sperrt fehlerhafte Tracks, unterstützt | 1698,– |
| 85 MB | Fast-Filing-System. Ubdateservice. | a. Anfr. |

Ideal als Monitoruntersatz, benötigt keine zusätzliche Stellfläche (33 cm B × 33 cm T × 6 cm H). Top Verarbeitung.

## Vesalia Filecard für Amiga 2000

| | | |
|---|---|---|
| 21 MB | Läuft unter Amiga-Dos auch ohne | 798,– |
| 31 MB | XT- und AT-Karte. Schneller Treiber. | 998,– |
| 47 MB | **3,5"-Harddisk.** Einfach in einen | 1289,– |
| 62 MB | Amiga-Slot stecken, fertig. | 1489,– |
| 85 MB | Auf Wunsch auch formatiert lieferbar. | a. Anfr. |

**Festplatten-Adapter für 3,5"-Harddisk im Amiga 2000**
Mit Trägerblech und Treibersoftware für 88 Harddisktypen.
Die Software sperrt fehlerhafte Tracks und legt Ersatztracks an, kompl. mit umfangreicher Anleitung. 99,–

## VESALIA AMIGA-ZUBEHÖR

**Boot-Selektor DFO - DF3** (elektronisch)
Es kann nun von allen Laufwerken gebootet werden.
Das interne Laufwerk ist abschaltbar. Einfacher Einbau, wird nur gesteckt, kein Löten erforderlich. **48,–**

| | |
|---|---|
| **Verbindungskabel Amiga an Monitor mit Scart** | 26,– |
| **Verbindungskabel Amiga an TV mit Scart** | 29,– |
| **Verbindungskabel Amiga an Multisync-Monitor** | 25,– |
| **Mausverlängerungskabel (3 Meter)** | 16,– |
| **Druckerkabel für Amiga 500/1000** | 21,– |
| **Druckerkabel für Amiga 2000** | 23,– |
| **64er Emulator-Ersatzkabel** | 16,– |

***Unser Renner: Top Verarbeitung, 18 Monate Garantie***
**3,5"-Laufwerk** superleise, mit durchgef. Bus bis DF3.
Abschaltbar, Metallgehäuse und Blende amigafarben. **239,–**

**3,5"-Laufwerk NEC 1036 a** mit durchgeführtem Bus.
Abschaltbar, Metallgehäuse und Blende amigafarben.
**6 Monate Garantie.** (Stückzahl begrenzt) **Im Versand** **209,–**

**3,5"-Laufwerk mit Trackdisplay NEC 1037**
Abschaltbar, Metallgehäuse und Blende amigafarben.
Mit durchgeführtem Bus bis DF3. **Im Versand** **299,–**

**3,5"-internes Laufwerk für Amiga 2000**
baugleich wie org. Commodore kompl. mit Zubehör 179,–

**Das Laufwerk für C-64-Aufsteiger (5,25"-Disketten)**
**5,25"-Laufwerk** 40/80 Track schaltbar und abschaltbar.
Metallgehäuse und Blende amigafarben. Mit unserem elektr. Bootselektor als DFO schaltbar, (nicht im Lieferumfang)
Durchgeführter Bus bis DF3. **298,–**

**Alle unsere Laufwerke sind kompatibel zur XT-Karte, zum Sidecar und zum PC1.**

**Unsere letzten NEC 1036 a zum Sonderpreis**
3,5" (org. verpacktes Rohlaufwerk) **159,–**

**512 KB-Karte für Amiga 500**
Mit akkugepufferter Uhr und abschaltbar. **329,–**

**512 KB-Karte** ohne Uhr 299,–

**2 MB-Box mit durchgeführtem Bus A 1000** 1198,–

**2 MB-Box für Amiga 500** 1198,–

***NEU NEU NEU NEU NEU NEU NEU NEU NEU NEU NEU NEU***
**2 MB-Box mit durchgeführtem Bus A 1000**
inkl. eingebauter Uhr und Kickstart 1.3 1398,–

**Sonderaktion: 500 Stück**
5,25" Laufwerk Philips 3113 org. verpackt 88,–
mit Schaltplan. Für CNC-Anlagen, Schneider, Apple- und Atari-Computer (80 Track einseitig)
Interface lieferbar. passendes Gehäuse 19,–

**Soundsampler für A 500/1000/2000**
Im Gehäuse, mit regelbarem Vorverstärker für Mikroaufnahmen, inkl. Software. 89,–

**Midiinterface für A 500/2000**
2 × Midi-Out, 1 × Midi-In, 1 × Midi-Thru. Mit Gehäuse und Anschlußkabel. 89,–

**Midiinterface für A 1000** 89,–

**Kickstart-Umschaltmodul** für drei Kickstartversionen 59,–

**Amiga-Clock 500/1000/2000**
kompl., mit Software. Anschluß an JOY-PORT 98,–

**+++ Händleranfragen erwünscht +++ Händleranfragen erwünscht +++**

**Betriebsferien (nur im Versand) vom 1. bis 31. Juli 89.**
**Vesalia Versand · 4230 Wesel · Magdalenenweg 4**
**Telefon 0281/65466 (8–17 Uhr) · Telefax 0281/64066**
**Ladenverkauf: 4230 Wesel · Kornmarkt 23**
**Montag–Freitag 9.30–18.00 Uhr**
**Samstag 9.30–13.00 Uhr**

*Vor kurzem hatten wir den neuen Flugsimulator im Test. Schon nach kurzer Zeit hat er viele Freunde und fanatische Flieger gezeugt. Doch das Programm ist nicht ohne. Neben gekonnter Flugakrobatik verlangt er vom User auch einige Geistesaktivitäten, die wir im folgenden Artikel mit einigen Tips herabsetzen möchten.*

VON HERBERT LORENZ

# Notlandung

## TIPS & TRICKS zu FALCON

### 'Duty Roster' & 'Sierra Hotel'

Um in das Cockpit der F-16 A (Fighting Falcon) zu gelangen, hat sich jeder Pilot erst einmal in den 'Duty Roster' (Dienstplan der z. Zt. aktiven Piloten) einzutragen. Solange ihm bei der Erfüllung der Missionen kein Unglück passiert, bleibt er hier mit seinem Namen bzw. seinem 'Call Sign', dem derzeitigen militärischen Rang und seiner Gesamtpunktzahl gespeichert.
Vorsicht! Wird sein Name überschrieben, verschwindet sein Spielstand für immer aus dem Duty Roster. Ebenso, wenn er den Status 'Active' durch ein Unglück verliert.

Anders verhält es sich mit dem 'Sierra Hotel', der 'Ewigen-Besten-Liste'. Hier bleiben die zehn besten Spieler, die jemals mit der Simulation gespielt haben, mit ihrem Rang in der Gesamtpunktzahl und ihrem derzeitigen Status [Active, Killed, POW (Kriegsgefangenschaft), Busted (unehrenhafte Entlassung), Missing in Action] gespeichert.
Ein guter Pilot bleibt also aufgrund seiner hohen Punktzahl auch nach einem evtl. Ausstieg aus dem 'Duty Roster' hier gespeichert.
Einzug hält nur, wer durch einen höheren Punktestand einen Vorgänger verdrängt. Jede weitere Manipulation entzieht sich den Möglichkeiten des Spielers.

### TIPS

Vorsicht bei der deutschen Version der Simulation. Im deutschen Handbuch sind nur neun (der insgesamt zwölf) Missionen erklärt.

Informationen zu folgenden Missionen fehlen in der deutschen Fassung des beiliegenden Handbuches:

** Rattlesnake Roundup*
*(Zerstörung von mindestens drei SAM-Stellungen im Norden)*

** Double Trouble*
*(Abschuß von zwei MIGs irgendwo im Feindgebiet)*

** Dragon's Tail*
*(Zerstörung der unteren Brücke im Westen)*

Beim Start sind die jeweiligen Zielkoordinaten schon im Autopiloten gespeichert. Vorsicht! Der Autopilot achtet nicht auf Berge und andere Hindernisse (ein Umstand, der auch im Luftkampf zum Verhängnis werden kann). Eine entsprechende Flughöhe sollte also schon vor der Aktivierung des Autopiloten erreicht sein. Wer gerne mit dem Autopiloten fliegt, kann nach der Erfüllung der Mission die Koordinaten des Heimatflughafens (D0) mit den 'Ö'- und 'Ä'-Tasten eingeben. Landen muß der Pilot aber nach wie vor allein.

Das Tüpfelchen auf dem 'i' ist aber die Möglichkeit, zwei Rechner miteinander zu koppeln und nun nicht gegen den Computer, sondern gegen einen anderen Spieler zu fliegen. Man muß sich nicht unbedingt bekämpfen. Das Üben von Formationsflügen hat auch seinen Reiz. Wer es dennoch darauf anlegt, seinen Spielpartner vom Himmel zu holen, und dabei mal den Kürzeren zieht, dem sei noch ein kleiner Trick verraten. Die Simulation weist (allerdings nur im On-Line-Modus) eine Lücke auf:
Wer getroffen zu Boden stürzt, dem kann durch das Gedrücktthalten der 'Help'-Taste und das Ausfahren des Fahrwerkes noch eine Notlandung gelingen. Kennt der Spielpartner diesen Trick nicht, braucht der Getroffene am Boden nur noch zu warten, bis sein 'erfolgreicher' Gegner Probleme mit dem Treibstoffvorrat bekommt. Wer auf derartige Winkelzüge des Spielpartners vorbereitet sein will, sollte sich zu Beginn noch eine Luft



<-> Boden - Rakete beim 'Sarge' besorgen.

Das Programm meldet im 'Zwei-Spieler'-Modus ob der Spielpartner den Autopiloten benutzt ('Your Opponent is using Autopilot'). Aufgrund der Schwachstelle des Autopiloten (s.o.) kann man ihn austricksen:

* *Sturzflug zum Boden und Abfangen in letzter Sekunde (wenn der Gegner direkt hinter einem ist) oder*

* *Sturzflug hinter einen Berg (wenn der Gegner noch etwas weiter entfernt ist) schöpfen das Unvermögen, Hindernisse zu erkennen, voll aus.*

Neben der direkten Rechnerkopplung besteht die Möglichkeit, per Modem gegeneinander zu spielen. Wer sich aber nur aus diesem Grunde erst ein Modem anschaffen will, dem sei dringend abgeraten. Die Geschwindigkeit der Simulation ist selbst bei einer Übertragungsrate von 2400 Baud unerträglich langsam. Erst 19200 Baud lassen wieder volle Spielfreude aufkommen. Zur direkten Rechnerkopplung hat sich obige 'Nullmodem'-Schaltung bestens bewährt.

| | | | |
|---|---|---|---|
| PIN 2 | <————> | PIN 3 | PIN 2 = TxD |
| PIN 3 | <————> | PIN 2 | PIN 3 = RxD |
| PIN 4 | <————> | PIN 5 | PIN 4 = RTS |
| PIN 5 | <————> | PIN 4 | PIN 5 = CTS |
| PIN 6 | <————> | PIN 20 | PIN 6 = DSR |
| PIN 20 | <————> | PIN 6 | PIN 7 = GND |
| PIN 7 | <————> | PIN 7 | PIN 11 = AUDO |
| PIN 11 | <————> | PIN 18 | PIN 18 = AUDI |
| PIN 18 | <————> | PIN 11 | PIN 20 = DTR |




Vorsicht vor PIN 9 und PIN 10! Hier liefert der AMIGA +12 Volt und -12 Volt. RECHNERZERSTÖRUNG !!!

Wer den Einsatz des Lötkolbens scheut, dem sei der RS232-Joker der Firma 'CONRAD-Electronic' empfohlen. Hier kann sich jeder seine eigene Schnittstelle durch einfaches Aufstecken von Verbindungskabeln (25 Stck. liegen bei) selbst konfigurieren. Genauso einfach lassen sich die Verbindungen auch wieder lösen! Durch das Zwischenstecken dieses Jokers und der Verwendung eines 25-adrigen seriellen Kabels (alle PINS 1:1 verbunden) ist man im Besitz einer universellen Schnittstellenverbindung für jeden Zweck!

## *Wünsche an das Programm*

Bei der Erfüllung seiner Missionen wird man immer von einer zweiten F-16, dem sogenannten 'Wing-Man' begleitet. Wer sich von ihm Hilfe erhofft, wird bitter enttäuscht. Er scheint einen privaten Friedensvertrag mit den MIG-Piloten geschlossen zu haben. Er wird weder angegriffen, noch ergreift er selbst irgendwelche Maßnahmen gegen die feindlichen Flugzeuge. Es wäre ja schon hilfreich, wenn er einem wenigstens nach Erfüllung der Mission im Landeanflug auf den Heimatflughafen den Rücken freihalten würde. Nichts ist ärgerlicher, als nach einer wirklich aufreibenden Luftschlacht mit den letzten Tropfen Treibstoff den Landeanflug zu beginnen und dann erneut von MIGs angegriffen zu werden.

Hat man sich zu Beginn der Mission für das Erscheinen von MIGs entschieden, ist das Programm unerbittlich und schickt nach jedem geglückten Abschuß eine weitere Maschine in die Schlacht. Die MIGs scheinen auch in der Luft von Nachschubproblemen verschont zu sein. Während man selbst ständig die Treibstoffanzeige und den Munitionsvorrat ängstlich im Auge behält, kämpft der MIG-Pilot scheinbar ohne derlei Limits. Im 'On-Line-Modus' besteht leider nur die Möglichkeit, gegeneinander zu kämpfen oder den Formationsflug zu üben. Es wäre wünschenswert, die Missionen gemeinsam fliegen zu können, so daß jeder Spieler des anderen 'Wing-Man' ist.

## Falcon Club

Begeisterte FALCON-Piloten können sich bei mir melden, da ich einen FALCON-Club ins Leben rufe.

# Frühlingspreise

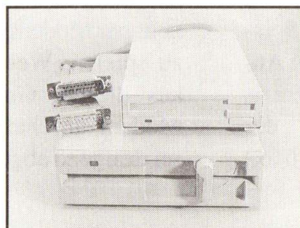
## Ein Preis- und Leistungsvergleich lohnt sich!

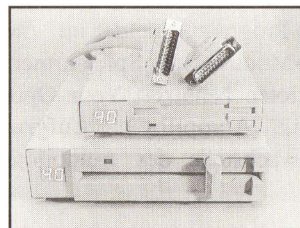
## Laufwerke

komplett anschlußfertig mit Kabel, amigafarbenes Metallgehäuse, abschaltbar, 3 ms Stepprate, Busdurchführung bis df3, DiskChange wird erkannt, 5,25"-TEAC-Laufwerke mit 40/80-Trackumschaltung.

| | |
|---|---|
| **SDN 3,5 – NEC 1037 A** | **219,-** |
| **SDN/DA 3,5**<br>*mit NEC 1037A und Busdurchführung* | **229,-** |
| **SDN 3,5 digital**<br>*mit Busdurchführung und Trackdisplay* | **259,-** |
| **SDN 3,5 – NEC 1036 A** | **219,-** |
| **SDN 3,5 intern**<br>*NEC 1036 A für A2000, Komplettkit* | **189,-** |
| **SDN 5,25 – TEAC** | **259,-** |
| **SDN/DA 5,25**<br>*mit TEAC FD 55 und Busdurchführung* | **269,-** |
| **SDN 5,25 digi**<br>*mit Busdurchführung und Trackdisplay* | **319,-** |

Alle 5,25"-TEAC-Drives können ohne Aufpreis auch mit 5,25"-NEC-Drives ausgeliefert werden.





## Festplatten

komplett anschlußfertig mit Controller, bereits formatiert und installiert, serienmäßige Auslieferung mit A.L.F. Wir verwenden ausschließlich Qualitätsfestplatten von SEAGATE und NEC, Fast-File-systemfähig.

| | | | |
|---|---|---|---|
| 30 MB A2000 | 745,- | 40 MB A2000 | 845,- |
| 60 MB A2000 | 1195,- | 50 MB FileCard | 1199,- |
| 30 MB A500 | 945,- | 40 MB A500 | 1045,- |
| 60 MB A500<br>mit SEAGATE-Harddisk | 1295,- | 60 MB A500<br>mit NEC 3,5"-Harddisk | 1495,- |
| 30 MB FileCard<br>für A2000, partitionierbar für MS-DOS und AMIGA-DOS | 895,- | 50 MB FileCard<br>für A2000, partitionierbar für MS-DOS und AMIGA-DOS | 1249,- |

**Stalter Computerbedarf GmbH**

*Gartenstraße 17 - 6670 St. Ingbert*



**06894/2012**

---

## 3,5" NN DISKS 2 DD

| | |
|---|---|
| 100 Stück | DM 199,95 |
| 50 Stück | DM 104,95 |
| 10 Stück | DM 21,95 |
| 1 Stück | DM 2,25 |

Fish, RPD, Faugh
Kickstart, Auge,
Panorama

**Publik Domain**

| | |
|---|---|
| 1 Stk | DM 4,50 |
| 10 Stk | DM 40,- |
| 50 Stk | DM 187,50 |
| 100 Stk | DM 350,- |

Amiga Publik Domain Handbuch
1-3 DM 49,-

video LOFT
HARD & SOFTware GmbH

Fiedlerstr. 22 - 32
D - 3500 Kassel
tel.: 0561 - 87 33 99 / 87 79 28

---

**fischer**
Hard & Software
Testen Sie uns

Leddinweg 14
3000 Hannover 61
05 11 / 57 23 58

## PUBLIC DOMAIN

ab 10 Stck. **2.75** incl. 2 DD 3,5" Disk

**Disketten sind etikettiert**

ab 10 Stck. **1.50** incl. 2 D 5,25" Disk

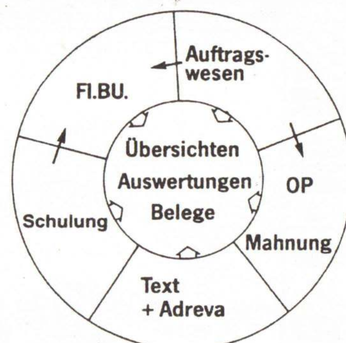
| | |
|---|---|
| Crazy Cars II | 75,– |
| Emerald Mines II | 37,– |
| Falcon F16 | 75,– |
| Popolus | 75,– |
| Test Drive II | 75,– |

!! Laufend Neuerscheinungen !!
Versandkosten NN + 8 DM, VK + 4 DM

---

## AMIGA-BUREAU

**ein Programm nach Ihren Wünschen**




**EDOTRONIK®**

D-8000 München 80, St.-Veit-Straße 70, ☎ 0 89 / 40 40 93

---

# Der neue Echtzeit-Videodigitizer SNAPSHOT!




Abbildungen wurden digitalisiert

Made in Germany

***1/50 Sekunde für ein professionelles Bild macht Schluß mit dem minutenlagen Warten.***

Telefonische Fachberatung ab 16 Uhr: (0431) 9 44 24
Händler-Anfragen erwünscht.

- Alle AMIGA-Grafikmodi werden unterstützt
- Auflösungen bis zu 704 x 552 Bildpunkten
- 16 / 32 / 46 Grautöne oder 4096 Farben
- 4 verschiedene Video-Eingänge

SNAPSHOT! Professional ..... **845,-**
RGB-Vorsatz für Farbbilder ........ **ab 125,-**
SNAPSHOT! STUDIO
(19 Zoll, Super-VHS, Farbe) ....... **1970,-**
2 Demo-Disketten ........................ **10,-**
Kostenlose Informationen anfordern!

**VIDEOTECHNIK DIEZEMANN** Dammstraße 42, 2300 Kiel 1




# ABO






ABO

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

**Heim Verlag**
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57




# Einzelheft- u. Monatsdisketten Bestellung






Einzelheft- u.
Disketten Service

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

**Heim Verlag**
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57




# Kontaktkarte






Kontaktkarte

Bitte Adresse der Firma, bei der Sie Informationen, oder etwas bestellen möchten, auf der rechten Seite eintragen → → → → → → → → → →

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ/Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte freimachen

Firma

Straße/Postfach

PLZ Ort

KICK START AMIGA ZEITSCHRIFT

# Abonnement

Ja, bitte senden Sie mir die Amiga-Computer Fachzeitschrift ab ____________
für mindestens 1 Jahr (11 Hefte) zum ermäßigsten Preis von jährlich DM 70,— frei Haus.
(Ausland: Nur gegen Scheck-Voreinsendung DM 90,— Normalpost.)
Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Name ____________

Vorname ____________

Straße/Nr. ____________

PLZ ____ Ort ____________

Gewünschte Zahlungsweise bitte ankreuzen

☐ Bequem und bargeldlos durch Bankeinzug

Konto-Nr. ____________ BLZ ____________

Institut ____________ Ort ____________

☐ Ein Verrechnungsscheck über DM______ liegt bei.

☐ Vorauskasse per Zahlung auf unser Postscheck-Konto Ffm, BLZ 500 100 60, Kto.-Nr. 5537-602

**Garantie:**
Diese Bestellung kann ich schriftlich innerhalb einer Woche (rechtzeitige Absendung genügt) widerrufen. Dies bestätige ich durch meine 2. Unterschrift.

Datum ____________ Unterschrift ____________

Datum ____________ Unterschrift ____________

KICK START AMIGA ZEITSCHRIFT

ABO

KICK START AMIGA ZEITSCHRIFT

# Einzelheft- u. Monatsdisketten Bestellung

KICKSTART können Sie direkt beim HEIM-VERLAG zum Einzelheft-Preis von DM 7,– (zuzüglich Gebühr für Porto und Verpackung) nachbestellen. Bearbeitung nur gegen beigefügten Scheck über den entsprechenden Betrag (keine Überweisung).

| Jan. | Febr. | März | April | Mai | Juni | Juli | Aug. | Sept. | Okt. | Nov. | Dez. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | 1987 = DM |
| | | | | | | | | | | | | 1988 = DM |
| | | | | | | | | | | | | 1989 = DM |

+ Gebühr für Porto u. Verp.

☐ Scheck in Höhe zus. DM liegt bei

## Disketten Service

Alle Programme, die in KICKSTART veröffentlicht wurden, sind auf Disketten erhältlich. Die Disketten enthalten die Programme von jeweils 2 KICKSTART-Ausgaben. Bestellen Sie durch ankreuzen die gewünschten Disketten

| Preis je Diskette 19,– DM | Juli/Aug. | Sept./Okt. | Nov./Dez. | Jan./Febr. | März/April | Mai/Juni | Juli/Aug. | Sept./Okt. |
|---|---|---|---|---|---|---|---|---|
| | 87 | 87 | 87 | 88 | 88 | 88 | 88 | 88 |
| | Nov./Dez. | Jan./Febr. | März/April | Mai/Juni | Juli/Aug. | Sept./Okt. | Nov./Dez. | Jan./Febr. |
| | 88 | 89 | 89 | 89 | 89 | 89 | 89 | 90 |

**Lieferung:** gegen beigefügten Scheck zuzügl. 5,– DM Versandkosten (unabhängig von bestellter Stückzahl)

KICK START AMIGA ZEITSCHRIFT

# Einzelheft- u. Monatsdisketten Bestellung

KICK START AMIGA ZEITSCHRIFT

☐ Ich bitte um weitere Informationen

☐ Ich gebe folgende Bestellung auf

in Bezug auf Ihre Anzeige in Kickstart Heft ________ Seite ____

| Menge | Produkt/Bestellnummer | DM | gesamt DM |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)



Abgesandt am:

KICK START AMIGA ZEITSCHRIFT

Firma:

Bemerkungen:




'TOP 12'
Mein Lieblingsspiel

'TOP 12'
Mein Lieblingsspiel

Wettbewerbsbedingungen siehe TOP 12 in diesem Heft

Bitte frei- machen

MAXON - Computer GmbH
Redaktion KICKSTART
'TOP 12'
Industriestraße 26

D-6236 Eschborn




Kurzmitteilung






Kurzmitteilung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Telefon

Postkarte

Bitte freimachen

MAXON Computer GmbH
KICKSTART Redaktion
Industriestraße 26

6236 Eschborn




PD Bestellung






PD Bestellung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte freimachen

MAXON Computer GmbH
KICKSTART Redaktion
Industriestraße 26

6236 Eschborn

KICK START
AMIGA ZEITSCHRIFT

'TOP 12'
Mein Lieblingsspiel

KICK START
AMIGA ZEITSCHRIFT

## Meine Meinung

Zu dem Artikel ____________ in Heft ____, Seite ____
hätte ich folgendes zu bemerken:

☐ Ich möchte Ihnen folgendes Programm zur Veröffentlichung anbieten: (Kurzbeschreibung, Sprache, Länge in Druckerseiten)

☐ Ich kann über folgendes Thema berichten: (Tips & Tricks am AMIGA, Hardware, Software, etc.)

☐ Ich möchte gerne Autor in der KICKSTART werden. Meine Fachgebiete: (z.B. LISP, Modula-2, Assembler...)

☐ Ich möchte, daß folgendes Public-Domain Programm in Ihre Sammlung aufgenommen wird.

☐ Sonstiges

Bei weiteren Angaben oder Fragen wenden Sie sich bitte schriftlich oder telefonisch an die Redaktion. Tel. 0 61 96/48 18 11



KICK START
AMIGA ZEITSCHRIFT

## Kurzmitteilung

KICK START
AMIGA ZEITSCHRIFT

## PUBLIC DOMAIN SERVICE

Ich bestelle folgende PD-Disketten:
(Siehe PD Service in dieser Ausgabe)

Zahlung erfolgt:

☐ per Scheck

☐ per Nachnahme
(nur Inland)
zuzüglich
DM 3,70 Nachnahmegebühr

Ab 5 Disketten Versandkosten frei

Je Diskette fügen Sie bitte einen Betrag von DM 8,— bei,
für Porto und Verpackung je Sendung DM 5,– (Ausland DM 10,–)

Datum

Unterschrift



KICK START
AMIGA ZEITSCHRIFT

## PD Bestellung

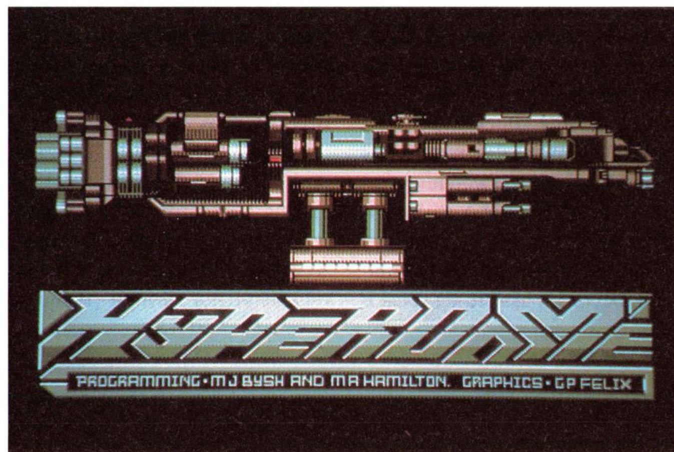
Datum

# DAS NEUESTE VOM SPIELEMARKT

von ddf, cpl und ak

## HYPERDOME

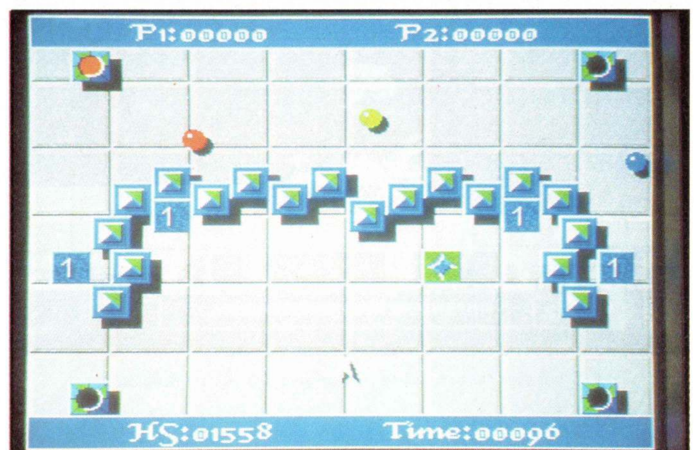
"THE FEDERATION" ist eine harte Truppe, fünf Jahre Training und nur die Besten sind gut genug zum Überleben. Am Ende der Ausbildung werden die Rekruten vor eine letzte Aufgabe gestellt - den HYPERDOME. Nur wenige sind geschickt und glücklich genug, diese letzte Aufgabe zu bewältigen - gehören Sie dazu? Dies ist ein seitlich scrollendes Schießspiel, bei dem Sie ein "Space Craft" steuern . Wie zu erwarten, ist der "Dome" voller Gegner, bereit und willens, Sie in Ihre Pixel zu zerlegen. Zerstörte Feinde hinterlassen statt einer Blutlache Extrawaffen wie Laser, Raketen, Doppelschuß und Schutzschild. Die neun Leben des scheinbar von einer Katze abstammenden Rekruten werden rasch weniger, aber nach einer Weile zeichnet sich ein Weg zur Lösung aller Probleme ab: feuern, ausweichen, feuern, feuern, feuern!!! In höheren Leveln nimmt die Zahl der Gegner zu, und es erscheint ein riesiger Alien, den man mehrmals treffen muß, bis er sich auflöst. Die Spielidee von Hyperdome ist so aufregend wie ein Fußballspiel mit Beteiligung der Frankfurter Eintracht, bedauerlicherweise liegt der Preis wesentlich höher. Die Spielfiguren bewegen sich allerdings mit vergleichbarer Langsamkeit über den Bildschirm. Zur Ehrenrettung muß man der Eintracht jedoch zugute halten, daß in den Pausen dort bessere Musik geboten wird.




*HYPERDOME ist ein weiteres Ballerspiel, das jedoch nicht überzeugen kann.*

## BEAM

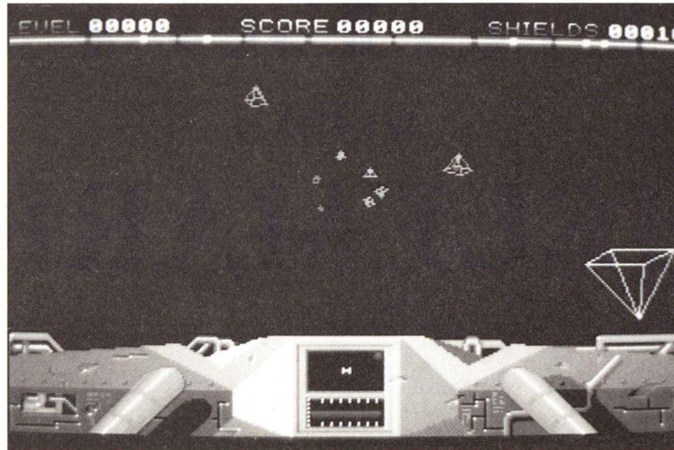
"...Des Master-Players coole Augen erblicken den 1.Level. Kerzengerade sitzt er vor dem Bildschirm, seine Hand hält locker den Joystick. Ganz öffnet sich sein Geist für das ihm dargebotene Bild. Intuition und Berechnung lassen ihm die Lösung wie ein Donnerschlag plötzlich klar werden. Die verwickelten Zusammenhänge der Objekte untereinander erscheinen ihm als Ganzes plötzlich so vertraut wie seine Hosentasche..." So beginnt wortwörtlich die deutsche Anleitung von BEAM, dem neuesten Spiel von MAGIC BYTES. Kein noch so brillanter Kommentar könnte alles, was es über dieses Programm zu sagen gibt, besser auf den Punkt bringen, als diese Selbstvorstellung, deshalb...




*Der Kommentar der Einleitung beschreibt das Spiel genau.*

# RAIDER

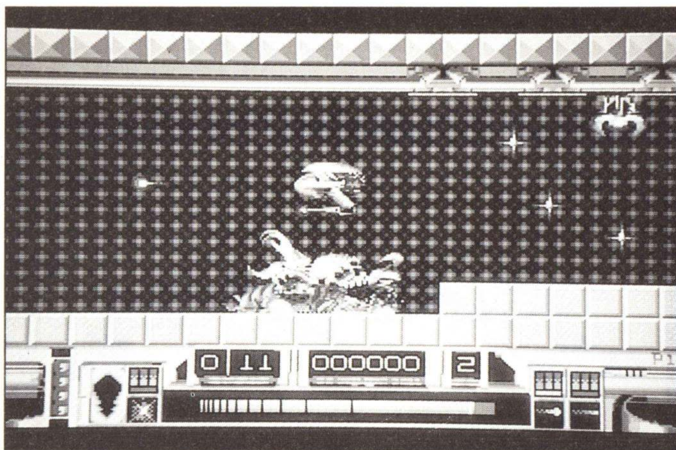
Wer das gute alte Billigspiel Thrust aus C64-Zeiten noch kennt (damals zehn Mark), hat eigentlich schon alles gesehen, was es über RAIDER zu sagen gibt - außer, daß die AMIGA-Variante mehr kostet und von einer Firma namens IMPRESSIONS stammt. Motto der Firma: SOFTWARE WITH STYLE - (h-) aha?! Ziel ist es, innerhalb einer bestimmten Zeit mittels eines winzigen Sprites andere Sprites in Mikroskopgröße einzusammeln, als einzelne Pixel getarnten Geschossen auszuweichen und am Ende des Levels ein Teil in den Laderaum des Dreiecks (Raumschif??) zu verfrachten, das etwa doppelt so groß ist wie jedes andere bewegte Objekt auf dem Bildschirm. Deshalb passen auch gleich vier Stück davon in den Hangar des nur halb so großen Dreiecks. Möglicherweise handelt es sich aber auch um eine vergrößerte Darstellung, da die Auflösung des AMIGA nach unten begrenzt ist und Objekte, die kleiner als ein Pixel sind, nicht mehr dargestellt werden können. Das Spiel wird komplett mit Keyboard gesteuert, neumodischer Schnickschnack wie Maus oder Joystick werden nicht benötigt. Die uns vorliegende Hintergrundstory entspricht dem Standardexemplar für intergalaktische Kurzanleitungen, das neuerdings immer häufiger Verwendung findet: Guter Herrscher A stirbt, böser Herrscher B übernimmt Macht, guter Held C muß D machen, bevor E passiert, da sonst F eintritt und G für immer verloren ist. In diesem Fall sind die Variablen wie folgt belegt: Galactic Emperor, Border Raiders, You, kill all turrets, powerplant appears, destroy, Star System.



*Frei nach dem Motto: Herrscher A stirbt, Herrscher B übernimmt die Macht, Held C muß alles retten (eingeschlossen den Käufer des Programms).*

Anmerkung des Verfassers:
"...Das Standardformular für intergalaktische Kurzanleitungen wurde übrigens auf dem ziemlich rückständigen dritten Planeten einer kleinen gelben Sonne der Kategorie fünf entwickelt. Seine Probephase wurde in vereinfachter Form in einer Fernsehsendung (rückständige zweidimensionale Wellenübertragung zur illusionären Darstellung von Pseudokommunikation) mit dem bezeichnenden Titel WETTEN - QUATSCH durchgeführt. Eine aus einer Beobachtermenge willkürliche Spezies hatte dabei die Aufgabe, eine Frage der Form: Wetten, daß Sie es nicht schaffen, A Personen auf die Bühne zu bringen, die B anbzw. dabeihaben und damit C machen, zu formulieren. Die zuschauende Masse hat allerdings den Schwindel schnell begriffen, und so wurde die Sendung bereits nach zwanzig Jahren wieder abgesetzt..." Auszug aus dem dritten Lexikon KURIOSITÄTEN MINDERENTWICKELTER PLANETEN, erschienen Standardjahr 1103 neogalaktischer Zeitrechnung als Klebebildsammlung in einem Müsliriegel.

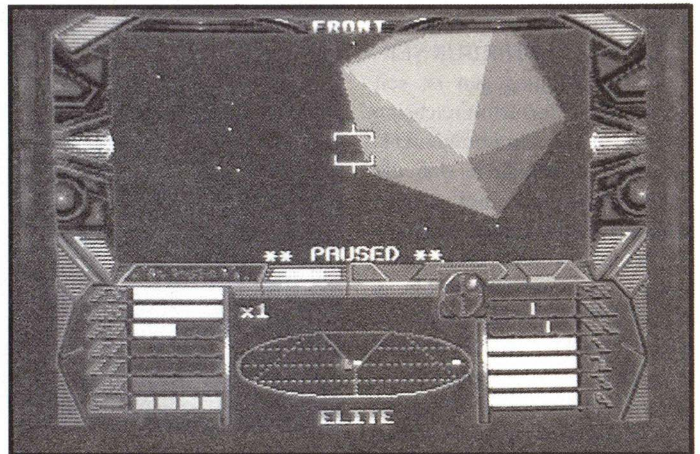
# JUG


*JUG hat eine Hintergrundstory, ansonsten das übliche Ballerspiel.*

Microdeals Neuerscheinung ist eine Mischung aus Labyrinth- und Ballerspiel. Das Ziel der extrem schwierigen Aufgabe ist es, ins Gehirn des lebenden Planeten Spiraeus vorzudringen, der sich in letzter Zeit recht merkwürdig benimmt. Dort sitzt ein Tumor, der für die Fehlfunktionen des Planeten die Verantwortung trägt. Sind Sie einmal in die Gänge des Planeten vorgedrungen, wird dessen körpereigenes Immunsystem aktiv und versucht, den Fremdling zu vernichten. Dies bedeutet, daß Sie von Anfang an mit Schwierigkeiten zu rechnen haben. Im Prinzip ist JUG ein horizontal scrollendes Schießspiel mit einer etwas einfallsreicheren Hintergrundstory als die meisten Vertreter dieser Art, sehr guter Graphik und einigen Extras. So kann sich der kleine Gleiter beispielsweise in drei verschiedenen Formen manifestieren. Auf den Spielverlauf hat dies keinen Einfluß. Möglicherweise fühlen Sie sich jedoch von einer der anderen Formen des Gleiters mehr angesprochen. Zu Beginn gibt es die übliche Laserkanone, und wie immer kann das "Schlachtschiff" mit aufzusammelnden Waffen nachgerüstet werden. Neu ist die Idee, daß beim Fliegen Energie verbraucht wird, die sich beim Fahren langsam wieder regeneriert. Insgesamt hebt sich JUG zwar positiv vom derzeitigen Durchschnitt ab, aber es ist halt nur "noch ein Ballerspielchen".

# TOP 12

1. (1) Elite




2. (2) Falcon
3. (3) Dungeon Master
4. (5) Holiday Maker
5. (7) Shanghai
6. (8) Ports Of Call
7. (10) Katakis
8. (4) Interceptor
9. (9) Bard's Tale II
10. (11) Sword Of Sodan
11. (12) Zak McKracken
12. (-) Populous

Die Spitzengruppe bleibt dicht beisammen und gibt keinen Platz frei. Anschluß gefunden hat das Artventure **HOLIDAY MAKER**, das sich seit einigen Monaten kontinuierlich nach oben arbeitet. Einziger Neueinstieg ist das "Götterspiel" **POPULOUS**, das wir diesmal vorstellen und auch gleich zum TOP 12-Tip des Monats gewählt haben. Dieses Spiel wird mit Sicherheit auch seinen Weg in die Spitze der Spielehitparade machen.

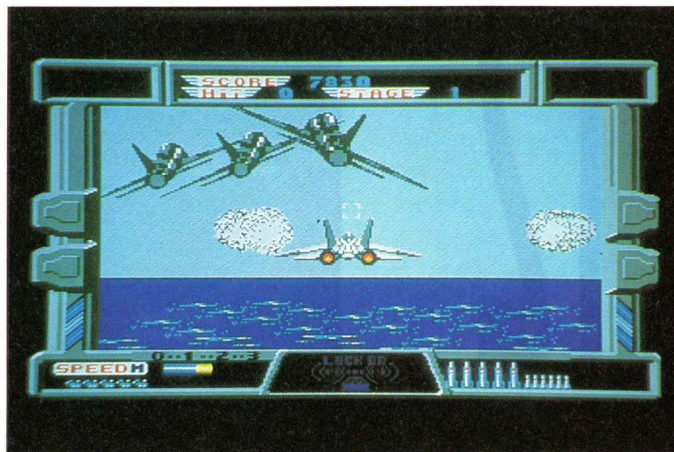
Mitmachen kann jeder, der die TOP 12-Postkarte (bitte keine andere!) ausfüllt und an uns zurückschickt. Der Rechtsweg ist dabei ausgeschlossen. Einsendeschluß ist der 30. April 1989. Karten, die nach diesem Termin eintreffen, werden im nächsten Monat berücksichtigt.

Wir gratulieren den Gewinnern dieses Monats, die jeweils einmal den **TOP 12-TIP** gewonnen haben:

Udo Pulm, Düsseldorf / Oliver Eichhorn, Obernburg / Dirk Tust, Kassel / Peter Sörgel, Nürnberg / Walter Ferenz, Neuß / Sven Kreutz, Lollar / Martin Ober, Lenz / Ralf Höhner, Bonn / Ingo Boker, Hille / Norman Kreffjargñow, Düsseldorf / Hermann Kuntsch, Dornstadt / Michael Länzer, Koblenz

# AFTERBURNER

Das 1988 erfolgreichste Spiel des japanischen Coin-Op-Giganten SEGA war AFTERBURNER. ACTIVISION hingegen ist seit Jahren eines der erfolgreichsten Softwarehäuser Großbritanniens. So ist es verständlich, daß die AMIGA-Gemeinde mit großen Erwartungen der Umsetzung harrte. Gestärkt wurden diese Hoffnungen auch von der Meldung, daß ACTIVISION das Team von ARGONAUT SOFTWARE (Starglider) als Programmierer gewinnen konnte. Der Erfolg des Automaten selbst wird verständlich, wenn man sich die technischen Daten von AFTERBURNER anschaut - mehr als drei Megabytes an 3D- Grafik, Sound und schneller Aktion. Obwohl das Konzept des Spiels recht einfach ist (ballern...), haben, neben der Tatsache, daß es teuer und äußerst schwierig war, noch andere Besonderheiten zu seinem Erfolg beigetragen. Die größte Attraktion war sicher die hydraulische, in alle Richtungen schwenkbare Pilotenkanzel, die dem Spieler das Gefühl gab, wirklich am Steuerknüppel der F14 zu sitzen. Mit reduziertem Speicherplatz und ohne die Möglichkeit, einen hydraulischen Sitz als "Gimmick" mitzuliefern (ein Poster ist das Höchste, was es kostenlos dazugibt), bringt uns die AMIGA-Version wieder auf den Boden der Tatsachen zurück.



*Afterburner kann die hohen Erwartungen keineswegs erfüllen, weder spielerisch noch grafisch.*

Beim Öffnen der Box fallen Ihnen zwei Disketten in die Hände. Aber unterliegen Sie nicht dem Glauben, Sie hätten zwei "Scheiben" voller AFTERBURNER-Code erworben, denn sehr rasch werden Sie feststellen, daß sich auf der zweiten Disk im Wesentlichen der Kopierschutz befindet. Das Spiel selbst ist, wie oben angedeutet, ein SHOOT-EM UP (im Wortsinn) mit mehr als 20 verschiedenen (?) Missionen, voller Luftkämpfe, Bomben, Raketen und Explosionen. Sie fliegen Ihre Tomcat durch einen endlosen Strom von Feinden, und die alles entscheidende Frage könnte von Darwin stammen: "Fressen oder gefressen werden?". Nichts Kompliziertes stört den Ablauf der Handlung, und so niedrig wie das Niveau des Spiels ist auch das der Programmierung. Die Steurerung des Flugzeues ist mäßig, und mit all den gegnerischen Flugzeugen und Raketen auf dem Bildschirm wird das Ganze so unübersichtlich, daß es fast unmöglich wird, zu sagen, wer hier auf wen feuert, und irgendwann weiß man selbst nicht mehr, ob man inmitten all dieser Explosionen auch selbst getroffen wurde. Kurz gesagt, das Spiel ist genauso unübersichtlich wie der letzte Satz.

Zu dem Illusionsverlust in Bezug auf den Unterhaltungswert gesellt sich die Enttäuschung über die Graphik. Sie bleibt weit hinter den Erwartungen zurück. Das einzige, was sie zu bieten hat, ist wohl ein Mangel an Details. Bäume, Büsche, Radartürme und Gebäude sind verschwommen, und die befremdenden Farbkombinationen tragen noch zur Verwirrung bei. Stellt man die großartigen Versprechungen der Hersteller den hohen Erwartungen gegenüber, die beim Publikum geweckt wurden, so ist AFTERBURNER schlicht eine Enttäuschung.
Aus diesem Grunde erhält Afterburner lediglich oder gerade noch die Bewertungsnote 3.

# SILKWORM

In Zusammenarbeit mit THE SALES CURVE präsentiert VIRGIN die Umsetzung eines kaum bekannten Automatenspieles - SILKWORM. Es mag nicht die aufsehenerregendste Lizenzvergabe für eine Konvertierung gewesen sein, aber es beinhaltet genügend Gemetzel und Zerstörung, um den Ansprüchen der Fans dieser Spielekategorie zu gefallen. Auslöser der Bildschirmschlacht war diesmal der Überfall böser Diktatoren auf ihr gepriesenes Vaterland (welches auch immer dies sei). Gerade als die Situation unhaltbar zu werden droht ... genau: fliegen, ausweichen, feuern, Reaktor erreichen und bumm - ewiger Friede (nein, kein GAU). Wie in den meisten Shoot-Em-Ups ist die Story recht bedeutungslos. Die Verpackung würde sonst wohl zu trist aussehen. Technische Daten: elf Levels, diverse Extrawaffen, zwei Spieler gleichzeitig, entweder mit Helikopter oder Jeep, Steuerung mit Maus, Joystick oder Keyboard, Scrolling und Graphik sind akzeptabel und auch der Sound entspricht dem Gesamteindruck des Spiels - die Jungs in den Schützengräben werden begeistert sein.



*SILKWORM bietet pure Action für sein Geld.*

# TOM & JERRY

## Wilde Hatz durchs Wohnzimmer

Die Comic-Helden Tom und Jerry fungieren im gleichnamigen Spiel des Gütersloher Softwarehauses Magic Bytes als Hauptdarsteller. Jerry ist eine kleine Maus, die gerne Käsestückchen stibitzt, während Kater Tom mit Genuß Mäuse wie Jerry verspeist. Normalerweise haust Jerry in einem Mauseloch. Da ist er vor Tom sicher. Doch immer, wenn er auf Käsesuche geht, versucht ihn der Kater zu packen. Ihre Aufgabe besteht darin, Jerry durch fünf Räume eines Hauses zu steuern und dabei alle herumliegenden Käsestückchen einzusammeln. Der Käse liegt dummerweise nicht auf dem Fußboden, sondern auf hohen Regalen herum. Da muß Jerry die Trampolinwirkung von Sesseln, Stühlen und anderen weichen Gegenständen ausnutzen, um in die Höhe zu gelangen. Eigentlich hat Jerry gegen Tom keine Chance. Der Kater kann auf die Regale springen und Jerry packen, wann er will. Doch glücklicherweise hat Jerry die Möglichkeit, Tom für einige Zeit außer Gefecht zu setzen. Heruntergeworfene Blumenvasen, Bowlingkugeln und andere Gegenstände, die auf Toms Kopf fallen, machen ihn für einige Zeit kampfunfähig. Dann kann Jerry in aller Ruhe Käse einsammeln. Hat er alle Käsestükke in einem Raum gefunden, kann er durch einen Mäusetunnel in den nächsten Raum gelangen. Im Tunnel kommen in 3D-Darstellung Bowlingkugeln und explodierende Geschenkpakete auf Jerry zu. Dann heißt es ausweichen oder darüberspringen. Bei einer Kollision geht nämlich Zeit verloren. Und davon hat Jerry nicht besonders viel. In 600 Sekunden muß er schließlich alle Käsestückchen in den fünf Räumen eingesammelt haben. Immer wenn Tom ihn zu fassen kriegt, bedeutet das einen Verlust von 30 Sekunden. Deshalb ist es wichtig, den nervenden Kater für kurze Zeit außer Gefecht zu setzen, da sonst einige Käsestükke unerreichbar sind. Die Grafik der wilden Hetzjagd ist auf dem AMIGA vortrefflich gelungen. Die Animationen von Tom erinnern ein bißchen an Zeichentrickfilm. Auch die Hintergründe und Einrichtungsgegenstände der einzelnen Räume können sich sehen lassen. Vom Sound war ich ein bißchen enttäuscht. Beim ersten Mal läßt




*Die Comic-Helden TOM & JERRY treiben ihr Unwesen auf dem AMIGA-Bildschirm.*

sich das "Ha, Ha ,Ha!" nach einem Zeitverlust ja noch ertragen, doch bekommt man diese digitalisierte Sprachausgabe andauernd zu hören, ist das Spiel nur noch mit Watte in den Ohren zu ertragen (Man kann natürlich auch den Soundregler leiser drehen!). Das gleiche gilt für die Jazzsounds, die während des Spielablaufs erklingen. Hat man sie einmal gehört, geht's noch. Aber andauernd... Da reißt man vor Ohrenschmerzen die Tapeten von den Wänden. Was den Spielspaß angeht, konnte mich "Tom & Jerry" hingegen voll überzeugen. Langeweile kommt so schnell nicht auf. Man steht schließlich andauernd unter Streß, weil Tom so gerne Mäuse mag. Der Preis von knapp 85 Mäusen ist dafür allerdings knapp an der Schmerzgrenze.

CBO

*Hersteller: Magic Bytes*
*Info: Ariolasoft*




# R-TYPE,

## Ballerei in Vollendung!




*Bei R-TYPE gilt es, 8 Level zu bewältigen, und alle bieten Action pur.*

Normalerweise landen Ballerspiele nach flüchtigem Ansehen sofort im Redaktionsmülleimer. Was nützt es Ihnen, liebe Kickstart-Leser, wenn wir über das 6789. Ballerspiel mit horizontalem Scrolling berichten. Doch "R-Type" können wir Ihnen nicht vorenthalten! Auch wenn man bei diesem Spiel tatsächlich nur bei horizontalem Scrolling über den Bildschirm fegt und Aliens abschießt, ist das AMIGA-"R-Type" etwas ganz Besonderes! Bei dieser Version handelt es sich um eine wirklich originalgetreue Umsetzung des Vorbilds aus der Spielhalle. Ziel des Spiels ist es, sich durch jeden Spielabschnitt zu ballern und dort dann ein Riesenmonster durch viele gezielte Schüsse zu vernichten. In jedem Level gibt es andere Bildschirmgegner, Musiken, Landschaften und Riesenmonster. Es greifen Bodenroboter an - kleine wendige Raumschiffe, fiese Mutanten und, und ... Jedes Level ist vollgepackt mit Action. Klar, daß man da schon Extrawaffen braucht, um sich der vielen Angreifer erwehren zu können. Ab und zu fliegen Satelliten mit Bonusgegenständen durch die Gegend. Dann heißt es schnell daraufballern und das Symbol einsammeln. Verschiedene Laserttypen und ein kleiner Satellit, der als Vorhut voranfliegt, dienen als effektive Extrawaffen. An einigen Stellen ist "R-Type" ganz schön schwierig, aber nie link. Mit Übung und der richtigen Taktik kommt man immer weiter. Das ist leider bei Ballerspielen nicht immer selbstverständlich. Oftmals werden solche Spiele auf die Schnelle produziert, so daß den Programmierern für Testspiele keine Zeit mehr bleibt. Konsequenz: Ein spielerisch unausgereiftes Produkt! "R-Type" hingegen ist perfekt! Herrliche Animationen der Gegner, stimmungsvolle Hintergrundgrafiken, exzellente Soundeffekte und passende Musik, eine ausgereifte Steuerung, butterweiches Scrolling, detaillierte Zeichnungen - einfach alles, was das Spielerherz begehrt. "R-Type" ist das beste Ballerspiel, das es derzeit für den AMIGA gibt.

CBO

*Hersteller: Electric Dreams*
*Info: Ariolasoft*



# Crazy Cars II - Fortsetzung folgt

War ein Titel auf dem hart umkämpften Softwaremarkt erfolgreich, kommt garantiert auch irgendwann eine Fortsetzung. Obwohl ich Crazy Cars von Titus als ziemlich langweiliges Spiel in Erinnerung hatte ("Lazy Cars"), schaute ich mir die Fortsetzung an: Als Agent in geheimer Mission jagt man mit über 200 Sachen über amerikanische Straßen, um rechtzeitig Beweise zur Zerschlagung einer Autoschieberbande irgendwohin zu bringen. Ein hartes Brot! Die Polizei ist nämlich ebenfalls in die Autoschieberei verwickelt. Doch auch die beflissenen und ehrlichen Bullizisten bereiten Schwierigkeiten: Straßensperren und zähe Polizeistreifen, die unentwegt hinter Ihnen herrasen, machen Ärger. Doch am meisten Kopfschmerzen bereitet die Geschwindigkeit Ihres Ferrari F40 und die gewöhnungsbedürftige Steuerung. Da muß man schon ein Fuchs an Maus oder Joystick sein, um auf der Straße zu bleiben. Mit einer Karte, die per Funktionstasten aufgerufen wird, müssen Sie Ihre Route planen. Dabei hilft Ihnen ein Radarwarnsystem, das vor Straßensperren warnt (sowas könnte ich für mein Auto gut gebrauchen!). Planung ist alles! Zeigt das Warnsystem eine Sperre an, gilt es, so schnell wie möglich die Straße zu wechseln. Achten Sie darauf, daß Sie dann auch in die richtige Richtung fahren. Vorsicht! Knallen Sie nicht in Telegrafenmasten oder Straßenbegrenzungen. Auch Zusammenstöße mit Polizeiautos sollten Sie vermeiden. Das kostet Zeit! Ich muß sagen, von "Crazy Cars II" war ich angenehm überrascht. Endlich passiert etwas in Titus' Autorennen. Im ersten Teil fuhr man auf einer langweiligen Strecke und überholte ab und zu andere Autos. Das war nicht die Erfüllung. Durch die Straßenkarten, das Radarwarnsystem und die herrliche Grafik kommt Pfiff ins Spiel. Fans von Rennspielen wie "Out Run" sollten mal einen Blick riskieren.




*Crazy Cars II kann als durchaus gelungen bezeichnet werden, jedoch verlangt die Steuerung des Ferraris viel Feingefühl.*

CBO

*Hersteller: Titus*
*Info: Leisuresoft*

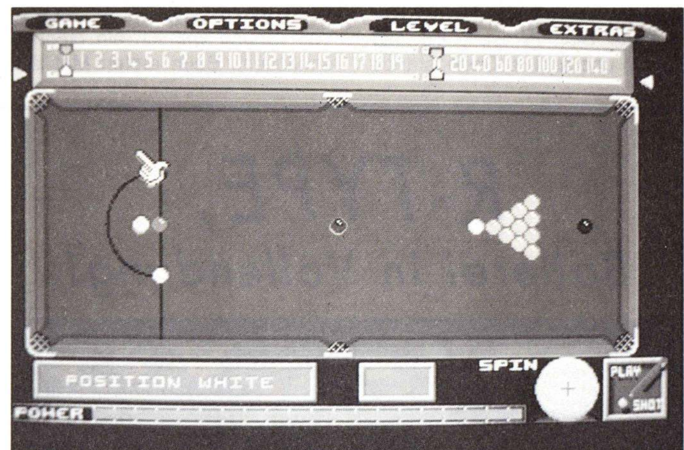



# World Snooker

## Billard für zu Hause

Billard ist ein schönes Spiel. Einige betreiben es sogar als Sport. AMIGA-User sind bisher nicht gerade mit guten Programmen dieser Kategorie verwöhnt worden. Das soll sich mit "Steve Davis World Snooker" ändern. Neben dem im Titel bereits angekündigten Snooker gibt es noch zwei Varianten des Pool-Spiels sowie zwei Versionen von Billard. Es sind also fünf verschiedene Spielarten auf den beiden Disketten. Jedes Spiel hat seine eigenen Gesetze. Die Regeln dazu sind kurz im Handbuch aufgeführt. Alle Optionen, die das Programm bietet, sind ausführlich in Englisch dokumentiert. Eine Option erlaubt z.B., sich in den verschiedenen Arten zu üben. Im Übungsmodus ist es auch möglich, Stöße zurückzunehmen oder zu wiederholen. Dabei wird der Spieler durch eine Voraussage-Option unterstützt. Mit ihrer Hilfe kann man sich ein Bild davon verschaffen, wie die Kugeln rollen. Jeder Stoß wird per Maus mit drei Parametern eingegeben: Richtung der Stoßkugel, Stärke des Stoßes und Effet. Mit dem Effet wird der Stoßkugel eine zusätzliche Drehung gegeben: Dadurch springt die Kugel von den Banden in einem anderen Winkel ab (also nicht Einfallswinkel=Ausfallswinkel). Diese Zusatzrotation wirkt sich nur an den Banden aus. In der Realität funktioniert dies auch bei Stößen auf eine Kugel. Der Effet bleibt bei der Voraussage unberücksichtigt. Schade! Einige Spiele werden nach Punkten bewertet. Wer als erster das Punkte-Limit erreicht hat, gewinnt. Diese Zahl läßt sich in weiten Grenzen einstellen. So kann man mal auf die Schnelle ein Spielchen wagen oder sich lange Partien liefern. Manchmal ist die Animation recht langsam: Trifft eine Kugel einen Pulk, dauert es einen Moment, bevor die Kugeln auseinanderfliegen. Es wäre besser gewesen, die Stöße vor der Ausführung zu berechnen. Soundmäßig hat das Programm auch Schwächen: Ein Zusammenprall mehrerer Kugeln klingt eher wie das Aufschlagen eines Tischtennisballes auf einer Platte. Dafür kommentiert Steve Davis höchstpersönlich als Schiedsrichter in verständlicher, digitalisierter Sprachausgabe das Geschehen. Die Mannigfaltigkeit an verschiedenen Optionen (wir können im Testbericht aus Platzgründen leider nicht auf alle eingehen) macht aus "Steve Davis World Snooker" eine echte Simulation. Wenn man von den beschriebenen Fehlern absieht, bleibt ein sehr gelungenes Billardspiel übrig.

CBO



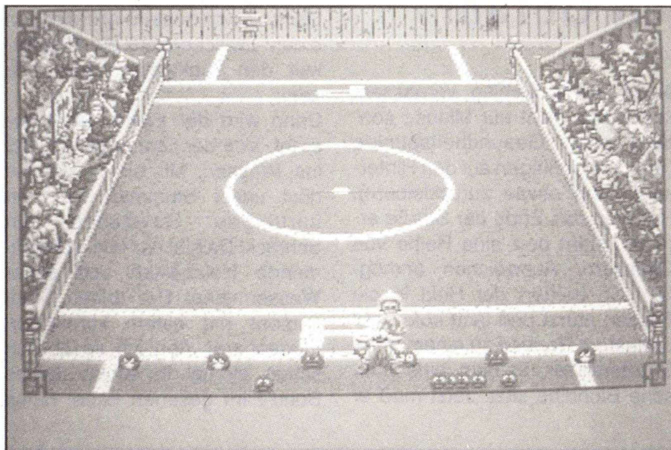

*Pool-Billard auf dem AMIGA - STEVE DAVIS WORLD SNOOKER macht es möglich.*

*Hersteller: CDS*
*Info: Leisuresoft*



# Grand Monster Slam

Irgendwo, auf einem fernen Planeten, treffen sich ein Haufen Monster und Fantasywesen, um einen sportlichen Wettkampf auszutragen: den "Grand Monster Slam". Dieser Sport ist straff durchorganisiert. Da gibt es wie beim Fußball mehrere Ligen. Sie sind ein kleiner Zwerg, der ganz unten anfängt, und sich in die erste Liga emporarbeiten muß, um dort schließlich Meister aller Grand Monster Slammer zu werden. In jeder Liga gibt es ein Viertel-, Halb- und ein Finalspiel, das man auch als Aufstiegsmatch bezeichnen könnte. Besiegt man alle drei Kontrahenten, muß man noch in einer Qualifikationsrunde kleine Fellkügelchen in hungrige Monstermäuler schießen. Erst wenn man zwei hineingeschossen hat, geht es in der nächsten Runde weiter. Ohne die Fellkügelchen, auch Belons genannt, geht es nicht. Sie sind nicht nur für die Qualifikationsrunden entscheidend. Auch für die normalen Ausscheidungsspiele sind sie von besonderer Bedeutung. Sie dienen nämlich als lebende Bälle. Und das ist kein leichter Job! Ziel eines Ausscheidungsspiels ist es nämlich, seine Belons auf die andere Seite zu kicken. Dort steht der Gegner und versucht ebenfalls, seine Belons loszuwerden. Das ist lustig anzuschauen! Ständig fliegen Belons hin und her. Um mehr Belons als der Gegner auf die andere Seite kicken zu können,



*Der Sportwettkampf der Monster bietet viele lustige Einfälle und gut animierte Sprites.*

muß man versuchen, mit Hilfe von gezielten Belonschüssen den Gegner anzuschießen. Der fällt dann für kurze Zeit um, und man kann in aller Ruhe ein paar Belons mehr hinüberkicken. Geben Sie acht, daß Sie die Fellkügelchen nicht ins Publikum fetzen. Das gibt eine Art Elfmeter als Strafe. Dann darf der Gegner einen Vogel in Ihre Richtung schießen. Wenn Sie den nicht auffangen, fliegen automatisch drei Belons mehr auf Ihre Seite. Äußerst unangenehm! Nach jedem gewonnenen Qualifikationsspiel läßt sich in einer Zwischenrunde der Score aufbessern: Mit einer Stange muß man wütende Belons abwehren. Hat man alle drei Partien und die Verfütterung der Belons überstanden, gelangt man in die zweite Liga. Dort gibt es wieder drei neue Gegner, die aber diesmal über gemeine Strategien verfügen. Während man die erste Spielstufe in relativ kurzer Zeit meistern kann, bleibt man in der zweiten Liga doch ganz schön lange stecken: Die Burschen sind echt zäh! Außerdem stehen mitten auf dem Spielfeld noch Mauern, an denen zu tiefliegende Belons wieder abprallen. Da kommt ganz schöner Streß auf! Grand Monster Slam macht wahnsinnigen Spaß. Sound (Chris Hülsbeck!) und Grafik sind Spitzenklasse! Die Motivation wird nur dadurch getrübt, daß man schon nach einigen Spielstunden das Game durchschaut und alle Überraschungen entdeckt hat. Andauernd Belons hin- und herkicken ist meiner Meinung nach auf die Dauer langweilig. Trotzdem kann man sich mit Grand Monster Slam länger beschäftigen als mit den meisten anderen AMIGA-Spielen. Viele Pluspunkte bekommt der Hersteller für die eigenständige Idee und die lustigen Einfälle, die in diesem Programm stecken. Ein Hoch auf Golden Goblins!

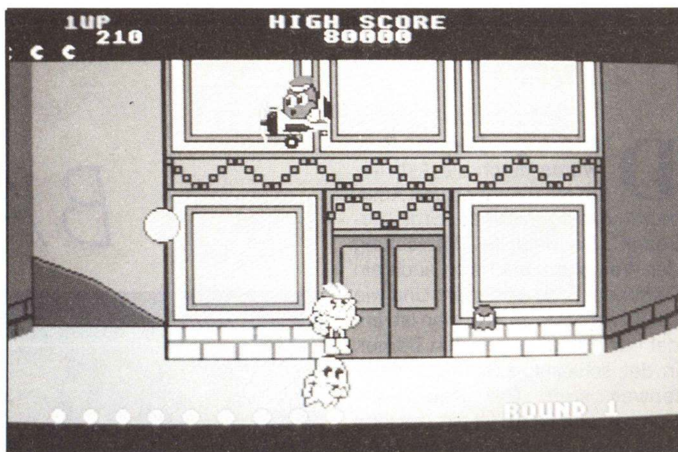
CBO

*Hersteller: Golden Goblins*
*Info: Rushware*




# Pac-Land

Pac Man ist wieder da! Diesmal mit Armen und Beinen! Richtig fein hat er sich gemacht, mit Mütze, Handschuhen und Stiefeln. Er trägt eine kleine Fee unter dem Hut und zieht mit ihr durch die Lande, von links nach rechts. Dabei wird er von seinen Erbfeinden verfolgt. Die Geister haben dazugelernt. Sie kommen jetzt in Flugzeugen angebraust und lassen Mini-Geister auf den Körnerfresser fallen. Oder sie kommen in Autos daher und versuchen, Pac Man zu überfahren. Für solche Fälle gibt es die guten alten Kraftpillen. Die Pillen stehen allerdings kurz vor dem Ende jeder Runde. Hat er eine Pille geschluckt, werden die Geister für einen Augenblick blau. Pac Man kann sie dann fressen. Die kleine gelbe Kreissäge (So sah er früher einmal aus, erinnern Sie sich?) wird dann noch von Hydranten, Baumstümpfen, Kakteen und anderen Kleinigkeiten aufgehalten. Mit dem Feuerknopf werden diese Hürden aber leicht übersprungen. Einige Gemeinheiten sind in die Landschaften eingebaut: Die Gebirge sind von Schluchten unterbrochen. Über diese Schluchten führen Holzstämme, die sich fahrstuhlartig auf- und abbewegen. Ein großer Swimming-Pool versperrt den Weg. Man kann ihn nur mit Hilfe eines Trampolins überspringen. Hier ist das Geschick gefordert, denn diese Stellen sind sehr schwer. Außerdem läuft Pac Man gegen die Zeit. Hat der Spieler eine Runde geschafft, erhält er die verbliebene Zeit als Punktebonus. Während einer Rund erscheinen Kirschen, die eingesammelt auch Punkte bringen. "Pac-Land" bringt frischen Wind in die schon recht eingefahrene Pac Man-Szene: Der Körnerfresser ist endlich aus seinem Labyrinth heraus. Dazu gibt es zeichentrickartige Grafik mit schnellem, horizontalem Scrolling

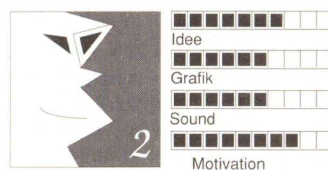



*Pac Man ist wieder da, jedoch ist er jetzt mit Mütze, Stiefeln und Handschuhen versehen.*

(das aber manchmal ruckelt) und lustige Musik. Besonders Kinder (auch große!) werden daran ihre Freude haben. Bis auf einige wirklich gemeine Stellen ist dieses Game toll zu spielen, - nicht nur für Pac Man-Enthusiasten.

CBO

*Hersteller: Grandslam*
*Info: Ariolasoft*



# DANGER FREAK

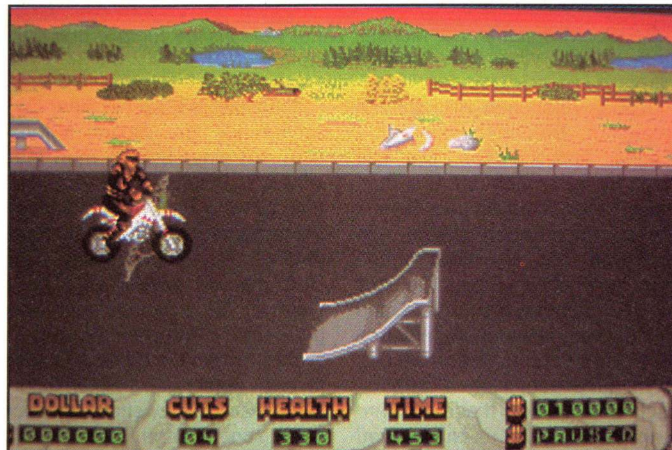
## Verrückt nach Gefahr

Filmstars halten bei Autokarambolagen, Stürzen von Hochhäusern und wilden Schlägereien nie selber ihre Knochen hin. Das erledigen Stuntmen für sie. Diese zähen Burschen sind speziell für derartige Aufgaben ausgebildet. "Rainbow Arts" widmet sich den heimlichen Stars Hollywoods im neuen AMIGA-Game "Danger Freak". In der Rolle eines Stuntman müssen Sie versuchen, bei drei verschiedenen Stunts möglichst viel Geld zu verdienen, und das möglichst ohne gesundheitliche Schäden. In der ersten gefährlichen Szene rasen Sie mit einem Motorrad in einem Kampf gegen die Zeit über eine schlecht asphaltierte Straße. Neben den Asphaltrissen sorgen Straßensperren, explodierende Minen und brennende Ölfässer für Schwierigkeiten. Karambolagen sollten vermieden werden, die Produzenten wollen schließlich perfekte Kunststücke sehen. Jeder Zusammenstoß mit einem Hindernis schlägt mit Minuspunkten zu Buche. Doch neben den Hindernissen gibt es noch andere Gemeinheiten auf der Strecke. Ab und zu sind Burschen mit schwingenden Baseballschlägern auf der Fahrbahn anzutreffen. Wenn die einem einen Schlag verpassen, kostet das nicht nur Minus-, sondern auch Gesundheitspunkte (Klar, bei Schlägen auf den Hinterkopf kann etwas zurückleiben!). Hat man das Ende der Straße erreicht, steht dort eine Reihe von Ölfässern. Ausweichen unmöglich! So brettert der Held in die Fässer, stürzt und läuft sofort auf Schusters Rappen zu einem roten Straßenflitzer, in dem eine hübsche Blondine sitzt. Jetzt muß er auf den Wagen springen, und nach kurzer Zeit läßt ein Hubschrauber eine Strickleiter herunter. Die klettert er hinauf, und schwupp geht's zum nächsten Stunt. Zuvor besteht allerdings in einem Zwischenspiel die Möglichkeit, den Punktestand aufzubessern.

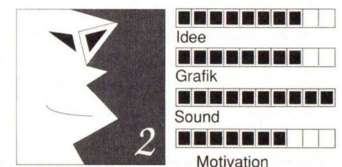
Dann wird der zweite Stunt gedreht. Von der Straße geht's jetzt ins Wasser. Mit einem Jetbike düst unser Stuntman über ein unruhiges Gewässer. Oh Schreck! Da gibt es Haie, schwimmende Holzbalken und sogar Wasserminen. Der dritten Stunt beginnt mit einem komischen Flugapparat, den ich so ähnlich schon einmal in einem James Bond-Film gesehen habe; damit geht es durch die Lüfte. Vorsicht vor Fahnenmasten, Flugsauriern (?), Raubvögeln und einer ferngesteuerten Rakete. Am Ende dieser Spielszene muß der Stuntman mit einem Fallschirm abspringen. Dann geht das Spiel wieder von vorne los, und man kann versuchen, noch mehr Punkte zu sammeln. Fazit: Vom Sound war ich beeindruckt. Soundmagier Chris Hülsbeck entlockt dem AMIGA tolle Musiken und fetzige Effekte. Auch grafisch ist das AMIGA-"Danger Freak" auf sehr hohem Niveau: Scrollender Hintergrund in drei Stufen, feine Animationen und schöne Zeichnungen. Spielerisch hat "Danger Freak" einiges zu bieten. Selbst wenn die drei Stunts alle über dasselbe Spielprinzip verfügen (bei horizontalem Scrolling Hindernissen ausweichen), bereitet das Spiel Spaß. Nach den Flops "Graffity Man", "Starball", "Reise zum Mittelpunkt der Erde" und "Australo Piticus" geht es anscheinend wieder aufwärts mit Rainbow Arts.

CBO

*Hersteller: Rainbow Arts*



*In die Rolle eines Stuntmans schlüpfen Sie bei dem Spiel Danger Freak.*

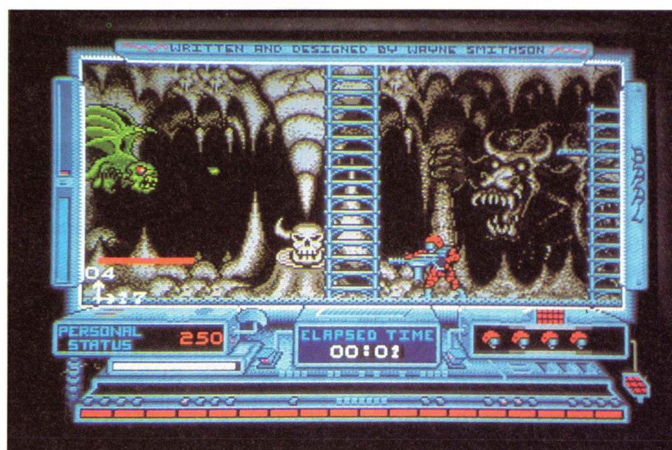



# BAAL

Der zornige Dämon Baal wurde durch die Aktivitäten einer Gruppe von Amateurarchäologen auf die Erde "gebeamt". Da Baal die Vernichtung der Welt plant, muß sich auch ein Retter der Erde einfinden. Und wie es nicht anders zu erwarten ist, findet der Kampf gegen den Dämon in der schleimig-schaurigen Höhlenwelt statt. Der Kampf mit Schwert und Dolch ist out - es lebe der Laser, also laß krachen, Kumpel...

Auf dem Weg durch verschiedene Levels und Hunderte von Szenarien müssen verschiedene Teile einer Kriegsmaschine gefunden werden, um diese für den "Endkampf" mit dem Monster parat zu haben. Die Suche erfordert gute Orientierung und gründliches Vorgehen, mithin viel Zeit. Baals Minen, Aliens und Plasma-Beamer sind auf dem Weg zu Baal durch Laserkanonen, die im Verlauf aufgepeppt werden können, und durch geschicktes Beamen zu erledigen, ehe dem Helden gegen Ende ein Raketentreibsatz auf dem Weg durch das Labyrinth hilft. Während er sich auf seinen langen, einsamen Weg begibt, schnell noch die Bewertung: Baal bringt nichts Neues, das aber im Standard. Das Spiel weiß durch einige gelungene Graphiken und durch gutes Scrolling zu gefallen. Dafür dürfen keine Ansprüche an Sound oder Originalität gestellt werden. Eine Empfehlung kann daher nur für Liebhaber dieses Spieltyps ausgesprochen werden.

*CPL*



*Dem bösen Dämon BAAL geht es ans Leder.*



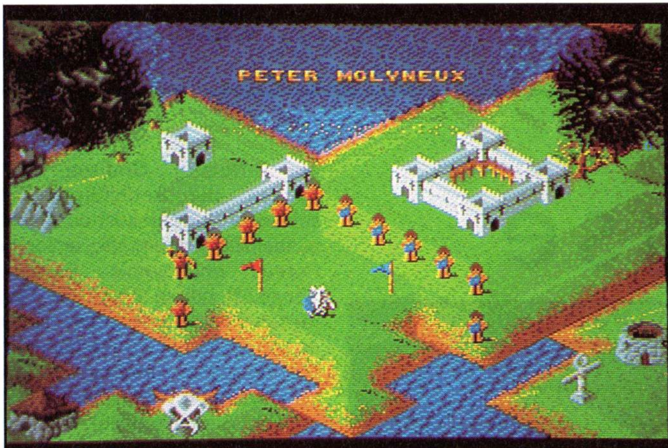
# POPULOUS - Die Welt zu Füßen

Mit Populous hat Bullfrog eine neue Dimension von Computerspielen eröffnet. Logisch, daß Bullfrogs Chefprogrammierer Peter Molyneux in diesem neuen Universum keine Giganten, sondern Gottheiten miteinander konkurrieren läßt. Populous ist dabei eine gelungene Mischung aus Strategie-, Action- und Glücksspiel. Der Gradmesser des Erfolges ist - wie sollte es bei Göttern anders sein - der Glaube der Bevölkerung an den "richtigen Gott".
Die Bevölkerung wird durch den Spieler indirekt gelenkt. Die Hilfsmittel sind dabei die in diesem Metier üblichen. Der Schöpfer richtet sich seine eigene Welt ein, indem er Berge, Täler und Flüsse entstehen, Wüsten zu Weiden werden läßt und dem Meer neues Land abnimmt. Damit sich die Bevölkerung fleißig mehre, kann er je nach Entwicklung der Population von der Hütte bis hin zum Schloß für die entsprechenden Unterkünfte sorgen. Je schneller die Häuser überbelegt sind, desto eher werden sich die Bewohner nach neuem Lebensraum umschauen und neues Territorium für sich und damit für ihren Gott erschließen.
Auf der anderen Seite der Beeinflussungsskala stehen die noch im Mittelalter anerkannten Gotteszeichen: Martialische Krieger, Naturkatastrophen wie Dürre oder Vulkanausbrüche führen zu unangenehmen Folgen für die Population. Repressionen sorgen bekanntlich dort, wo Gutes nicht mehr wirkt, oft noch für den angestrebten Erfolg...
Allein die große Variationsvielfalt - immerhin besteht die "Welt" aus 10.000 Szenarien-, die detailliert ausgearbeiteten Graphiken und der plastische Sound sind begeisternd und zeugen von der professionellen, sechsmonatigen Arbeit des Teams von Bullfrog.
Die neue Dimension von Populous beginnt auch jenseits dieser Bewertung. Die aufwendige perspektivische Graphik erfordert für den Betrieb mit zwei Spielern zwei Rechner. Diese werden einfach über die serielle Schnittstelle verbunden. Zur weiteren Steigerung will Bullfrog im Spätjahr den Erweiterungssatz Populous Plus mit zusätzlichen Szenarien und Computerstrategien herausbringen. Ebenfalls in Arbeit ist eine Ausgabe, die es bis zu 16 Spielern im Netzwerkverbund erlaubt, gegeneinander zu spielen.

CPL



Bei Populous tritt der Spieler in die Fußstapfen eines Gottes und muß die Entwicklung eines Volkes lenken.



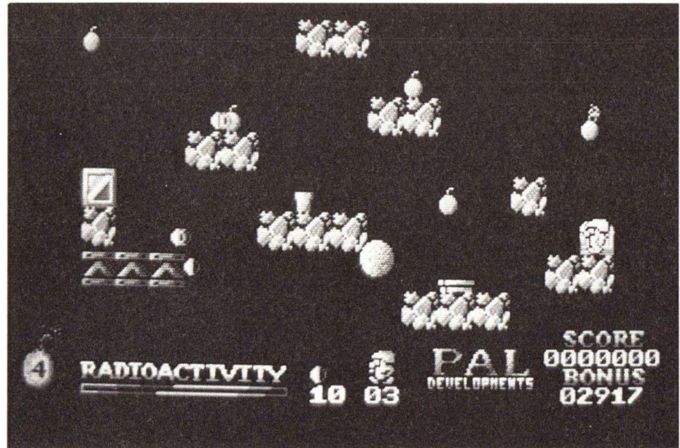

Nicht nur die Spielhandlung ist göttlich, auch die Grafik kann man so bezeichnen.



# Bomb Fusion

Ein Profi muß her! Terroristen haben Bomben in einem Atomkraftwerk gelegt. Obendrein versagt das Computersystem, das für die Wiederaufbereitung der verbrauchten Brennstäbe zuständig ist. Aufgabe des "Bomb Fusion"-Spielers ist es, den atomaren Treibstoff einzusammeln und ihn dann in Transportkisten zu verfrachten. Brennstäbe werden aus einem Rohr ausgestoßen und irren ziellos umher. Der Spieler sammelt sie durch Berührung ein. Wenn die Zündschnur einer Bombe brennt, heißt es, schnell dorthin zu springen. Durch eine einfache Berührung wird die Bombe entschärft. Gelingt es Ihnen nicht, rechtzeitig eine aktive Bombe zu erreichen, macht es "BUMM". Der Bildschirm blitzt nur einmal kurz auf. Drei Explosionen kosten ein Bildschirmleben. Am unteren Bildrand befindet sich ein Bombensymbol, das die Zeit bis zum Knall anzeigt. Daneben ist ein Balken angebracht, an dem sich ablesen läßt, wie stark der Held radioaktiv verseucht ist. Steht diese Anzeige auf Maximum, sind Sie ebenfalls ein Leben los. Doch es gibt noch andere Gemeinheiten: Berührt der Held eine große Kugel, die sich über den Bildschirm bewegt, steigt die radioaktive Verseuchung. Außerdem nimmt sie ihm die eingefangenen Brennstäbe wieder ab. Gegen die Kugel ist kein Kraut gewachsen. Man kann ihr höchstens ausweichen. Mit zehn Brennelementen beginnt das Spiel. Pro Level steigert sich diese Zahl um zwei. Bei Null wird es Zeit, das Level zu verlassen. Das geht nämlich nicht automatisch. Der Profi muß selbst zum Ausgang. Dieser ist als Gitter dargestellt. Waren Sie schnell genug, gibt es noch einen Punktebonus. Ohne Hilfe von Trampolins kann der Profi nicht alle Stellen des Screens erreichen. Springt er auf eines, kann er auch höher gelegene Plattformen erreichen. Fällt der Spieler unten aus dem Bildschirm, kommt er oben wieder herein. So gelangt man zwar auf alle Plattformen, jedoch kostet diese Methode sehr viel Zeit. "Bomb Fusion" ist grafisch schlicht aufgemacht. Es gibt nur wenige verschiedene Plattformtypen. Eine schrille Melodie begleitet das Spiel. Sie ist glücklicherweise auch abschaltbar. "Bomb Fusion" ist zwar kein Renner, aber immer noch besser als Mastertronics andere Neuerscheinung "Chase". Ein schwacher Trost!

CBO




*Weder grafisch noch spielerisch kann Bomb Fusion überzeugen.*

*Hersteller: Mastertronic*
*Info: Leisuresoft*




# Chase

Böse Buben entführten eine Weltraumprinzessin namens Chardonnay. Jetzt muß ein Held her, der das holde Mädel befreit. Mit einem Raumschiff nehmen Sie die Verfolgungsjagd auf. Soweit zur Hintergrundgeschichte von "Chase", Mastertronic's neuestem Billigspiel für den AMIGA. Besagte Verfolgungsjagd besteht aus vier Phasen: Zuerst geht es durch einen Asteroidensturm. Jeder Zusammenprall sorgt für den Verlust eines Schutzschilds. Zehn Schilde haben Sie zu Beginn. Die Entführer fliegen in blauen Raumschiffen vor Ihnen her. Fünf Treffer brauchen Sie, um ein Schiff zu sprengen. Ein Radarschirm am unteren Bildrand erleichtert das Zielen. Sie können auch ein Visier in das Cockpitfenster einblenden. Die zweite Phase ist eine Art Slalomfahren. Es gilt, zwischen zwei Säulen hindurchzufahren. Zehn solcher Slalomtore stehen dazu bereit. Haben Sie die hinter sich gelassen, beginnt die dritte Phase: ein Tunnel. Er wird ähnlich wie beim alten C64-Klassiker 'Magic Carpet' durch herannahende Quadrate dargestellt. Ein einziger Crash mit den Rändern sorgt dafür, daß es in Phase vier weitergeht (unlogisch!): In Star Wars-Manier fliegen Sie dort durch einen Korridor und ballern mit Hilfe des Fadenkreuzes feindliche Schiffe ab. Doch Vorsicht! Rote Rechtecke versperren den Weg. Riskante Ausweichmanöver sind angesagt. Ist auch diese Phase geschafft, geht das Spiel wieder von vorne los - mit dem Unterschied, daß in einigen Stufen die Angreifer andere Formen besitzen. Gesteuert wird das Raumschiff per Maus. Mit der linken Maustaste feuert man, mit der rechten wird beschleunigt und gebremst. An einigen Stellen reagiert die Steuerung sehr schwerfällig (das gibt Abzüge in der B-Note!!!). Grafisch stellt "Chase" durchschnittliche bis schlechte Softwarekost dar: nicht ausgefüllte Vektorgrafik, einfache Formen und wenige Farben. Sound ist nur in Form von Geräuscheffekten vorhanden. Doch es gibt noch mehr zu bemängeln: Die deutsche Anleitung ist ein Witz, was den Informationsgehalt angeht. Sie beschreibt lediglich die Kontrollmöglichkeiten. Auch wenn "Chase" zum günstigen Preis von ca 35.- DM angeboten wird, lohnt sich ein Kauf nicht. Selbst für ein Billigspiel ist "Chase" äußerst schwach.

CBO



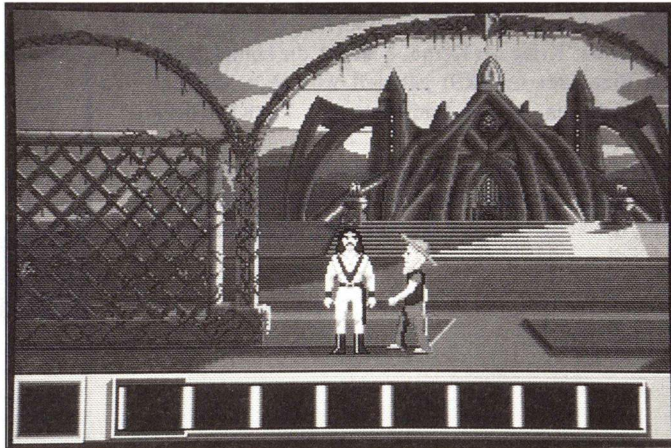

*Mit Chase hat sich Mastertronic wirklich ein Ding geleistet, mit einer derart schlechten Programmierung kann man wirklich keinen AMIGA-User hinterm Offen hervorlocken.*

*Hersteller: Mastertronic*
*Info: Leisuresoft*



# THE KRISTAL

## Auf der Suche nach dem Kristall



Man merkt es THE KRISTAL, dem neuen Spiel von Addictive Games, sofort an, daß die Handlung an ein Theaterstück angelehnt ist ("The Kristal of Kronos" von Rodney Wyatt). Obwohl die Rahmenhandlung dem Spielverlauf untergeordnet ist, ergibt sich eine dichte Atmosphäre, die den Spieler immer wieder aufs Neue motiviert. THE KRISTAL ist insgesamt wie ein Film aufgebaut. Nach dem Titelbild folgt eine ungefähr 20 Sekunden lange Sprachausgabe, die das Beste ist, was ich auf dem AMIGA jemals gehört habe. Wer schon einmal Orson Wells auf der Neuabmischung von Alan Parsons' "Tales Of Mystery And Imagination" reden gehört hat, kann sich ungefähr vorstellen, wie die Sprachausgabe klingt. Wow! Ebenso besitzt THE KRISTAL einen eigenen Soundtrack, der insgesamt über dem Durchschnitt liegt. Die Graphik und die Animation der Spielcharaktere sind - wie sollte es auch anders sein - vom Feinsten. THE KRISTAL wird auf vier Disketten ausgeliefert, was dem Umfang des Spiels gerecht wird.

Soviel zur technischen Seite. In THE KRISTAL übernimmt der Spieler die Rolle des Dancis Frake, der auszieht, um den Kristall von Kronos zu finden. Die Spielhandlung findet auf acht Planeten statt, die die Tochtersterne Kree und Ma umkreisen. Auf diesen Schauplätzen findet Dancis nützliche Objekte, kann aber auch mit zufällig auftauchenden Personen reden, um weitere Informationen über den Kristall zu bekommen. Weiterhin gibt es die Möglichkeit, einzelne zu einem Schwertkampf herauszufordern. Dem gesamten Spiel haftet eine üppige Portion englischer Humor an. Betrachtet man THE KRISTAL als Ganzes, so ist es wirklich ein gelungenes Spiel, dem man die Detailfreude der Programmierer anmerkt. Doch wo sooo viel Licht ist, ist auch Schatten. Der dunkelste Schatten weit und breit ist der Diskettenzugriff, der wirklich gähnend langsam ist. Außerdem verbreitet er einen Lärm, der die Ohren beleidigt und der Mechanik des Laufwerks den Rest gibt. Vielleicht ist es deshalb auch eine Form englischen Humors, daß in der Laufschrift vor dem Spiel von einem eigens entwickelten FastLoad gesprochen wird. Weitere Schwächen von THE KRISTAL liegen in der Texteingabe und im Parser. Dieser versteht zwar ohne weiteres das Wort ****, bezeichnet man aber das vorher eingegebene Subjekt als "it", gibt es Verständnisschwierigkeiten. Auch - aber das ist wirklich der letzte Kritikpunkt - ist der Schwertkampf zu schwer geraten, so daß man am Anfang kaum eine Chance hat. Alles in allem bleibt jedoch ein monumentales Werk übrig, an dem sich folgende Spiele dieses Genres werden messen müssen.

*Sven Stillich*




# Heroes of the Lance



*Heroes of the Lance wird sicherlich viele Rollenspielfans begeistern können.*

Der Cataclysm traf die Einwohner scheinbar völlig unvorbereitet, der Weltuntergang schien nahe. Der Zorn der Götter über die Bevölkerung und besonders über den Führer der Kirche, der mit unverschämten Forderungen die Götter beleidigte, war gewaltig. Städte wurden vernichtet, und die Landschaft von Krymm veränderte sich durch zahlreiche Explosionen. Doch es kam noch viel schlimmer, denn in ihrer Wut wandten sich die Bewohner von den Göttern ab und vergaßen sie über die Jahre. Hier beginnt Ihre Aufgabe. Mit einer Gruppe von Abenteurern, den 'Companions of The Lance', müssen Sie das verlorengegangene Wissen über die Götter wiederherstellen. Der einzig mögliche Weg ist, die 'Disks of Mishkal' zu finden, denn nur mit ihrem Wissen kann man den Glauben an die Götter wiederherstellen. Dies ist deshalb so wichtig, weil nur mit ihnen die Göttin des Bösen davon abgehalten werden kann, ganz Krymm in ihre Hand zu bekommen. Doch mit den Disks kann die Göttin der Finsternis besiegt!?, in die Abyss verbannt werden. Natürlich liegen die Scheiben nicht für jeden zugänglich irgendwo herum, sondern sind tief in den Ruinen von Xak Tsaroth in der Höhle eines riesigen, schwarzen Drachen verborgen. 'Heroes of the Lance' wurde von den beliebten 'Dragonlance' Romanen und Rollenspielmodulen des AD&D Rollenspielsystems auf den Computer übertragen. Es handelt sich allerdings nicht um ein reines Rollenspiel, vielmehr überwiegt der Actionanteil. Sie steuern mit dem Joystick immer einen der acht Charakter. Bei einem beliebigen Tastendruck erscheint ein Menü, mit dem Sie verschiedene Aktionen ausführen können, wie z.B. Zaubern oder Gegenstände aufnehmen. Am Anfang sind die Monster noch harmlos, was sich aber schnell ändert, wenn Sie tiefer in die Ruinen eindringen und ihnen die ersten 'Babydrachen' über den Weg laufen. Hier ist Flucht manchmal nicht die schlechteste Lösung, auch wenn das nicht sehr Heldenhaft ist. Haben Sie aber erst einmal die Trolle, Söldner und etlichen anderen Monster auf welche Art auch immer überwunden, kommen Sie in die Höhle des Löwen, der sich in diesem Fall als schwarzer Riesendrache wirklich 'ätzend' bemerkbar macht, hier hilft auch Wegrennen nichts.

Die Grafik ist recht gut gelungen, obwohl sie mich stark an ST-Grafik erinnert. Für jeden Charakter steht ein anderes, großes Sprite zur Verfügung, das gleiche gilt für die Monster. 'Heroes of The Lance' bietet viel Atmosphäre, die sich mit den Romanen fast unbegrenzt steigern läßt, und liefert endlich mal eine wirklich ausgereifte Hintergrundstory.

*Michael Jahn*



# KICKSTART PU

Auch in diesem Monat gibt es wieder eine Menge neuer Spieledisketten, die sich alle nicht hinter kommerziellen Versionen zu verstecken brauchen. Als interessante Anwendung gibt es diesmal die Simulation CORE WARS, auch als "Krieg der Kerne" bekannt, die komplett mit Programmiersprache (REDCODE) und Simulator (MARS) geliefert wird. Für Modula-Programmierer gibt es zwei gefüllte Disketten, wobei besonders das MAKE-Utility Aufmerksamkeit verdient.
Programmierer werden sich sicherlich für die neue Diskette von COMMODORE zum IFF-Standard interessieren, denn hier findet man die wichtigen Dokumentationen und Routinen. Den Abschluß bildet wieder die Utility-Diskette, auf der die zur Zeit interessantesten Hilfsprogramme vertreten sind.

Bis zum nächsten Mal

Markus Nerding

# DIE NEUHEITEN

## KICK 161: SPIELE



**MORIA:** Irren Sie gerne durch dunkle Verliese, und wollen Sie unheimliche Mächte und ihre Wesen bekämpfen, dann ist dieses Spiel das richtige. Zu Beginn müssen Sie im Dorf Hilfsmittel, Waffen und magische Sprüche erwerben, und dann können Sie sich in das Abenteuer stürzen.
Mit schöner Grafik, einfacher Benutzerführung (Menüs), vielen Optionen, die die Spielbarkeit erhöhen und dem AMIGA gerecht werden. Benötigt 1MB Speicher.

## KICK 162: SPIELE

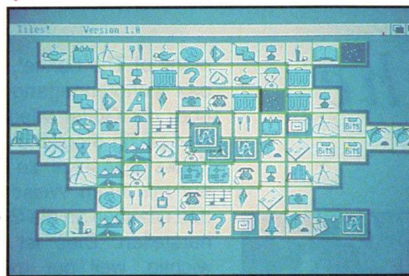



**PACMAN87:** eine gelungene Pacman-Variante mit einigen neuen Features wie z.B. Feuerfallen, Messern, Elektrofeldern und einiges mehr. Drei Schwierigkeitsgrade, HighScore-Tabelle, Sound, Joysticksteuerung.

## KICK 163: SPIELE

**WORLD:** Ein Textadventure ähnlich den Infocom-Adventures Planetfall und Starcross. Guter Wortschatz (ca. 500) und umfangreiche Texte (130 KB). Mit Source-Code in C.

## KICK 164: SPIELE

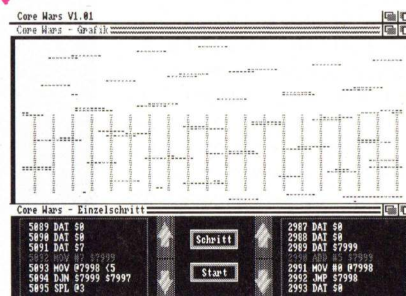



**TILES:** Eine schöne Version des unter dem Namen Shanghai bekannt gewordenen Spiels. Aus einem Steinestapel müssen solche mit gleichen Symbolen gefunden werden, die gleichzeitig freiliegen, also nicht von anderen Steinen umgeben sind. Wird es gelingen, den ganzen Stapel aufzulösen? Ein spannendes Taktikspiel für alle Spielefans.

## KICK 165: SOUNDS

**SOUND QUEST I:** Einige sehr gute Sounds (Blue Monday, Rocking, Bomb the Kaes, Slice Me Nice, Tron, Hunters Moon, That's all), die von ACS mit dem Soundtracker erstellt wurden.

## KICK 166: SIMULATIONEN




Diese Version ist auch unter dem Namen "Krieg der Kerne" bekannt. Dabei bekämpfen sich zwei Programme innerhalb des Rechners. Sie testen Speicherzellen, umkreisen den Gegner oder greifen ihn direkt an. Komplettes System zum Erstellen der Programme (Assemblersprache REDCODE) und Abwickeln der Simulation (MARS). Mit umfangreichen deutschen Anleitungen (200 KB) und 12 Beispielprogrammen. CORE WARS ist Shareware. Autoren: Roger Meier und Patric Hofmann von Unicorn Systems.

## KICK 167: MODULA (AMOK 15)

**Kurve:** leistungsfähiges Kurvendiskussionsprogramm (Ableitungen, Integrale, Nullstellen, Extrema, Wendepunkte usw.)
**Detach:** endpositioniert Icons
**Originale:** Rückübersetzung einiger M2Amiga-Module
**Seafarer's Manual:** Programme und Lösungen
**Calc:** ähnlich EVAL von WB 1.3

## KICK 168: MODULA (AMOK 16)

**M2MAKE:** unverzichtbares Make-Utility für jeden ernsthaften Modula-Programmierer.
**MODLIST:** formatierter Ausdruck von Modula-Listings mit hervorgehobenen Schlüsselwörtern.
**ROTATEIFF:** rotiert ein IFF-Bild um die Horizontale.
**MULTISELECT:** Mehrfachauswahl mit dem rechten Mausknopf anstelle von Shift.
Außerdem Updates von: MemSystem V1.3, AVL V1.2, LIST V1.1, QUEUE V1.2

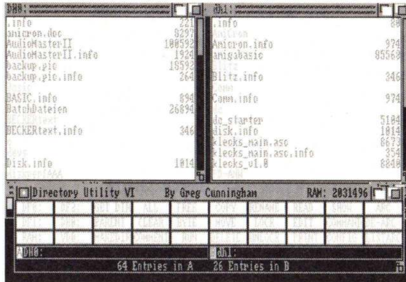
## KICK 169: UTILITIES

**IFF-STANDARD:** Dies ist die neue Diskette zum IFF (Interface File Format)-Standard von ECA und COMMODORE, die anläßlich der DevCon 89 veröffentlicht wurde. Hier sind die Standards (ILBM, FTXT, SMUS, 8SVX) und neue Kennungen spezieller Anwendungen (ANIM, MIDI, WORD, RGB4 u.v.a.) festgelegt, an die sich jeder Softwareentwickler halten sollte, um kompatible Programme zu erhalten. Mit den neuesten Versionen von SHOWILBM, DISPLAY, SCREENSAVE und MORE.

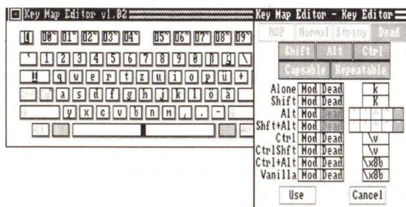
## KICK 170: UTILITIES

**DISKPERF:** Ein Benchmark-Programm für Diskettenlaufwerke, Festplatten und andere Speichermedien. Enthält umfangreiche Tests für Schreib- und Lesezugriffe und gibt eine Liste der Daten auf. Einige Daten für verschiedene Medien und Rechner (VAX unter UNIX) sind bereits enthalten. Das neue Fast File-System der WB 1.3 wird berücksichtigt. Source in C.

# BLIC DOMAIN

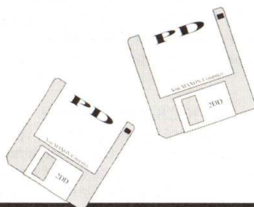


**DIRUTIL VI:** Ein sehr komfortabler und leistungsfähiger Ersatz für das CLI. Alle wichtigen Befehle und Funktionen können mit der Maus ausgeführt werden.



**KEYMAPED:** Ein Editor für die Tastaturbelegung der SetMap-Dateien. Sehr komfortabel und einfach mit der Maus zu bedienen. Mit Source in C.

**DIFFDIR:** Vergleicht zwei Directories. Sehr nützlich für einen Aktualitätscheck. Source in C.



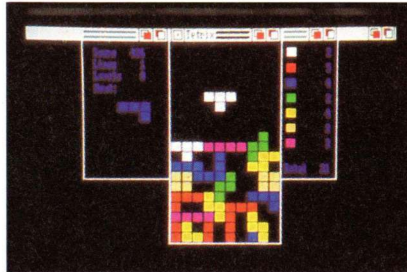

ANRUF GENÜGT
06196 / 481811

## KICK 152: SPIELE



**CASTLE OF DOOM**: Ort der Handlung ist Sherwood Forest in England im 12. Jahrhundert, und die Aufgabe ist an die Robin Hood-Sage angelehnt. Ein schönes Adventurespiel mit Grafik. Für Anfänger geeignet, mausgesteuert, englisch.

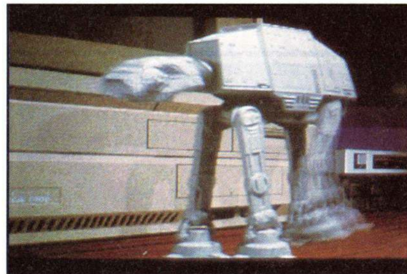
## KICK 153: SPIELE



**TETRIX**: Herabfallende Formen müssen so gedreht werden, daß sie zusammenpassen. Trotz dieser einfachen Spielidee ist es eines der besten Spiele für den AMIGA. Wer einmal damit angefangen hat, den wird das Spiel so schnell nicht mehr loslassen.

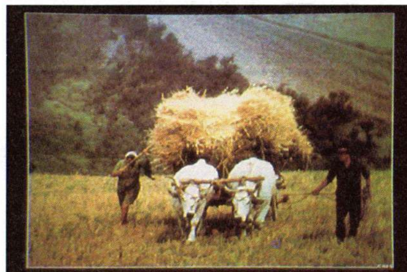
**FIGHT**: Eine Variante des Pong-Spiels. Zweispielermodus, Source in C (interessante Effekte für eigene Programme). *Autor: Matthias Kühn*

## KICK 154/155: GRAFIK



**WALKER**: Auf zwei Disketten befindet sich diese Megademo. Sie benötigt mindestens 2MB Speicher und zeigt einen Walker (bekannt aus 'Krieg der Sterne'), der vor einem AMIGA umhermarschiert. Phantastische Demo mit sehr guter Animation und bombastischem Sound.

## KICK 156: GRAFIK

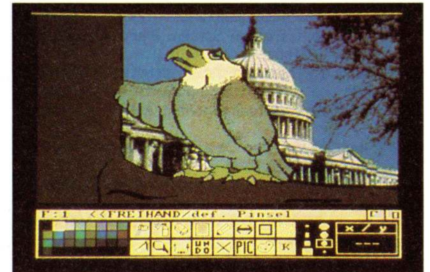


Acht digitalisiert Bilder in der Auflösung Lores/Interlace/HAM. Sehr schöne Motive und gelungene Digitalisierung. Inklusive **vollautomatischem** Projektor mit verschiedenen Überblendeffekten(!).

## KICK 157: SOUNDS

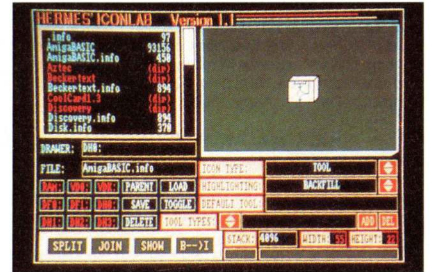
Nachdem der **Soundtracker** unter Freaks schon lange zum Standard geworden ist, zeigen die Sounds dieser Diskette eindrucksvoll dessen Leistung. *Soundmaster: Holger Blaschka*

## KICK 158: ANWENDUNGEN



**KLECKS V1.0**: Ein Malprogramm mit vielen Funktionen, guten Fähigkeiten und komfortabler Bedienung. Obwohl es in AmigaBasic geschrieben ist, erreicht es beachtliche Geschwindigkeiten (ein Lob an den Autor). Einige Optionen: Dateiauswahlbox, Füllmustereditor, Pal, Hires, Overscan, Fonts, Pinsel, Vergrößern, Drehen, Spiegeln, Kopieren uva. Benötigt mind. 1 MB. *Autor: Thomas Auf der Straße*

## KICK 159: UTILITIES



**HERMES' ICONLAB V1.1**: Das beste Hilfsprogramm zum Bearbeiten von Icons. Es erlaubt das Wählen von 'Icon Type', 'Default Tool', 'Tool Types' und des Stacks. Außerdem können Icons zusammengefügt, getrennt und aus Brushes (z.B. von Deluxe Paint) erzeugt werden. Sehr komfortable Bedienung.

## KICK 160: UTILITIES

**VSCREEN**: Sensationell, denn dieses Programm vergrößert den WB-Screen auf einen beliebigen Wert (begrenzt vom ChipMEM). Wenn man dann die Maus an den Rand bewegt, scrollt der Screen sanft weiter und weiter.

**KLICKIT**: Dieses Programm ermöglicht es, CLI-Befehle und Batch-Dateien von der Workbench aus zu starten. KLICKIT generiert dazu ein Startprogramm, das auf Wunsch ein Fenster für eine Eingabe/Ausgabe öffnet. Einfache Maussteuerung.
*Autor: Udo Marquardt.*

**WICON**: Schafft Ordnung auf dem Bildschirm. Jedes Fenster kann von nun an auf Knopfdruck bis auf die Menüleiste verkleinert und entsprechend auch wieder vergrößert werden.




**Fortsetzung umseitig !**

## KICK 151-127

### KICK 151: SPIELE
**BROKER 1.0**: Das Handeln mit Aktien ist die Aufgabe der bis zu vier Spieler. Beim Kaufen und Verkaufen sind Bonus, Steuern, Optionen, Dividenden, Zuschießen, Kontozinsen und Kredite zu berücksichtigen. Spiel in zwei Versionen (deutsch und englisch), mausgesteuert.
*Autor: Andreas Page*

### KICK 150: UTILITIES
**TASKDRIVE**: ermöglicht ein bequemes Ansehen und Ändern der Abarbeitungsprioritäten jedes Prozesses/ Tasks im System. Eigener Screen, Source in Lattice C. Autor: Rudolf Gerlinger (Österreich)
**IMAGCALC**: ein Taschenrechner, der mit komplexen Zahlen arbeitet und deshalb z.B. die Wurzel einer negativen Zahl in Imaginär- und Realteil zerlegt. Autor: Peter Schaefer
**CODE**: ein Chiffrier- und Dechiffrier-Programm mit Paßwortschutz. Autor: K.J. Wolf
**SHC (Screen-Hard-Copy)**: Ein sehr schnelles (Assembler) Druckprogramm für eine Bitplane eines Screens. Für STAR LC-10, NL-10, EPSON LX-800 und Kompatible. Mit Source. Autor: Karsten Kalweit

### KICK 149: MODULA (AMOK 14)
**ARP**: M2AMIGA-Implementation der arp.library, die das BCPL-DOS durch leistungsfähigere Befehle ersetzt. **TIMERSUPPORT**: erleichtert die Benutzung des Timer-Deviced

### KICK 148: MODULA (AMOK 13)
**COMPCONTROL**: erleichtert die Bedienung von Editor, Compiler und Linker. **ROWS**: generischer Datentyp, während der Laufzeit dimensionierbare Arrays. **WITH**: ein neuer CLI-Befehl. **FLIPMOUSE**: invertiert die Richtung der Mausbewegung. **WBSHADOW**: Schatten für die Workbench. **WBPIC**: bringt ein Bild auf die Workbench.

### KICK 147: ANWENDUNGEN
**M.S.TEXT**: Eine leistungsfähige Textverarbeitung mit den wichtigsten Funktionen. Einfache Bedienung über Menüs oder Tastensequenzen, Blockfunktionen (markieren, kopieren, löschen, laden, speichern, drucken), eine Dateiauswahlbox und die schnelle Scrollgeschwindigkeit zeichnen sind die wichtigsten Merkmale. Quellcode in AZTEC C, komplett in Deutsch. Autor: Martin Steppler

### KICK 146: ANWENDUNGEN
**NOSCOM-MATCH 1.1**: Hierbei handelt es sich um ein Auswertungsprogramm für Wettkämpfe des offiziellen nationalen Schießsports. Es ist sehr komfortabel und fast gänzlich mit der Maus zu bedienen. Es erleichtert Darstellung, Auswertung und Ausdruck der Ergebnislisten. Eine Version 1.2 ist in Vorbereitung und wird vom Autor angeboten. Komplett in Deutsch, mit ausführlicher Dokumentation; benötigt 1MB RAM. Autor: Norbert Süßdorf

### KICK 145: ANWENDUNGEN
**GIROMAN V3.00**: Ein sehr leistungsfäiges Girokondtoverwaltungsprogramm, das vom Autor schon seit 1980 auf einem PET eingesetzt wird. Das Programm ist sehr komfortabel und einfach zu bedienen. Zu den wichtigsten Funktionen zählen: Einträge suchen, Kontostand eines betimmten Tages abfragen, monatliche Festposten (Daueraufträge usw.) erledigen, Ausdruck mit Saldoangabe und manuelle Buchungen (hinzufügen, ändern, löschen). Mit Anleitung, komplett in Deutsch, Source in Aztec C. Autor: Sönke Netzband

### KICK 144: GRAFIK
**AMIGAPAINT V1.5**: ein deutsches Malprogramm mit vielen Optionen (Freihand, Punkte, Linie, Rechteck, Block, Fläche, Kreis, Füllen, Radiergummi, Text, Spraydose, Füllmuster und Lupe). Es arbeitet im Lores-Modus mit 320x200x5 und 32 Farben. Auf der Diskette ist das Listing in AmigaBASIC (etwa 1240 Programmzeilen) enthalten.

### KICK 143: MUSIK
**WIZARD OF SOUND**: ein Musikprogramm zum Erstellen von Musikstücken nach Notenvorlage oder zumKomponieren. Einige Funktionen im Überblick: bis zu 8 Instrumente gleichzeitig, 4 Stimmen, verarbeitet gesamplete Instrumente im IFF-Format (8SVX / 5 Oktaven), Inplay-Modus (Spielen mit Begleitmusik), Spzialfunktionen (Crescendo, Accelerando, Ritardand, Fermate, Triole usw.), autom. Taktreorganisation, Ausdruck der Notation, Player-Programm (läuft im Hintergrund), multitaskingfähig, komplette Maussteuerung, Help-Funktion, dt. Anleitung. Autor: Jörg Schließer

### KICK 142: SPIELE
**GAMBLER**: Ein allgemeines Spielsystem, das Strategiespiele wie Dame, Vier gewinnt, Tic-Tac-Toe u.ä. spielen kann. Mit diesem Programm errang Guido Appenzeller einen Preis bei der Regionalausscheidung von "Jugend forscht" 1988/89.
**BALLY**: Das Spielprinzip ist sicherlich bekannt - 75%einer Fläche müssen umrandet und damit ausgefüllt werden, nur darf man sich nicht von den herumschwirrenden Punkten erwischen lassen. Sehr schnell, Joystick erforderlich! Autor: Oliver Wagner

### KICK 141: SPIELE
**PARANOID**: eine Breakout-Variante mit High-Score-Liste und Level-Editor. Mit dem Leveleditor kann man sich auf einfache Weise beliebige Levels gestalten. Auf der Diskette sind bereits 24 Levels vorhanden. Mit gutem Sound. Autor: Felix Wente

**Aus Platzgründen können leider nicht mehr alle PD-Disketten aufgeführt werden. Sollten Sie keine ältere KICKSTART zur Hand haben, dann können Sie die Liste auch bei uns anfordern. Senden Sie dafür einen adressierten und mit DM 1.- frankierten DIN A5 Rückumschlag an folgende Adresse:**

**MAXON Computer GmbH**
**KICKSTART PD-Liste**
**Postfach 55 69**
**6236 Eschborn**

## Machen Sie mit!

Möchten auch Sie selbstgeschriebene Programme der Allgemeinheit zur Verfügung stellen, so schicken Sie sie uns einfach zu (bitte mit Dokumentation auf Diskette!).

Als kleine Anerkennung können Sie sich dafür 5 Disketten aus unserem PD-Service auswählen.

*MAXON-Computer GmbH*
*KICKSTART-Redaktion*
*PD-Einsendung*
*Industriestraße 26*
*6236 Eschborn*

### KICK 140: UTILITIES
**RAMCOPY**: Kopierprogramm für ein Laufwerk und mind. 1MB Speicher
**DISKSALV**: rettet gelöschte Dateien
**SPUDCLOCK**: Uhrzeitansage
**MACHII**: Maustool, Hotkeys, Popcli usw.
**MULTICALC**: RPN-Taschenrechner
**MRPRINT**: Druckprogramm für Dateien, mit vielen Optionen
**AFTTERM**: Terminalprogramm
**MFIX**: patcht das Kopierprogramm MARAUDER und steigert damit die Geschwindigkeit erheblich

### KICK 138 & 139: UTILITIES
Auf dieser und der KICKPD 139 befinden sich die Sourcecodes der meisten Programme von Matt Dillion, der sicherlich jedem PD-Anwender bekannt ist. Im einzelnen sind dies: CONFIG V1.00, CLOCK V1.00, DME V1.31, DMOUSE V1.10, BACKUP V2.01, SUPLIB, LIBREF, DRES V1, DASM V2.11, FTOHEX, FILES V1.2, SHELL V2.10, FINDIT V1.00, LIBS V1.00, SCAT, ADDCR, REMCR, CMP.

### KICK 137: SPRACHEN
Für alle Programmiersprachenbegeisterten ist hier eine kleine Fundgrube. **SCHEME**: ein Dialekt ähnlich Lisp, **PILOT**: eine Umsetzung dieser Sprache für den AMIGA, **ADA:** Syntaxkontrolle für ADA, mit YACC-Source, **PERL**: Interpreter

### KICK 136: ANWENDUNGEN
**AMIGASPELL**: Wörterbuch (Wortschatz: 9085 engl. Wörter) zum Korrigieren von (ASCII)Texten. Falsche Wörter können ausgedruckt, im Text korrigiert (der Satzzusammenhang wird angezeigt!) oder übernommen werden. Schnell (Assembler) und speichereffektiv. Es kann leicht ein deutsches oder hessisches Wörterbuch erstellt werden.
**NAG**: der absolute, blinkende, schreiende, singende Terminkalender, damit Sie nie einen Geburtstag oder eine Fete oder vergessen. Sehr komfortable Eingabe, viele Möglichkeiten, immer griffbereit.
**HP11**: HP11C-Calculator mit allen Finessen.

### KICK 135: GRAFIK
**MANDEL 2.1**: Mandelbrotgenerator für repititive Muster. Source in AmigaBASIC. *Autor: Michael*

### KICK 134: BILDER
Viele gute Grafiken (SexyRobot, Roland SH-101, Sailboat, Max, Ferrari GTO, Dali, ST), Cartoons (Batman, Garfield) und digitalisierte Bilder (Supercollage von Marilyn, Mona Lisa, Alien).

### KICK 133: GRAFIK-SHOW
**CRAZY-MOTOR-SHOW**: gut gezeichnete Sportwagen und eigene Konstruktionen mit Angabe der technische Daten und interessanten Hintergrundinformationen. Mit digitalisierter Musik. 1MB Speicher erforderlich. *Autofan: I.K.*

### KICK 132: SOUNDS
**TGB SOUND DISK NO.1**: Fünf Musikstücke (erstellt mit Soundtracker) werden vor dem Hintergrund von HAM-Bildern abgespielt. *Komponist: Thomas-Georg Brüggemann.*

### KICK 131: SPIELE
**TENNIS**: Tennisspiel für zwei Spieler mit Trainingsmodus, Grund-linien- und Netzspiel. Die Schlagstärke und Richtung können mit dem Joystick bestimmt werden. Erfordert 1MB Speicher. *Autor: Christian Dreke.*

### KICK 130: ANWENDUNGEN
**SUPERLIGA**: Fußballergebnisverwaltung mit 577 Mannschaften aus 23 Ländern und 34 Ligen. Über 10000 Ergebnisse können gespeichert werden. Das Eingeben gestaltet sich sehr einfach. Aus den bisherigen Ergebnissen kann eine Prognose (Regressinsanalyse) für kommende Spiel und die Meisterschaft erstellt werden. Autor: Rolf Morlock

### KICK 129: SOUNDS
**Neue Kompositionen** von Andreas Starr aus Gronau, erstellt mit Sonix. Effektvolle Stereostücke mit Player.

### KICK 128: SOUNDS
**Scores, Drumkits und Soundfiles** für Dynamic Drums von New Wave Software.

### KICK 127: GRAFIK
**Mandelbrot-Show II**: dies ist der zweite Teil der Reise durch die Welt der Mandelbrotgrafiken. Autor: R. Beck

## VERSANDBEDINGUNGEN:

Um einen schnellen und problemlosen Versand zu gewährleisten, beachten Sie bitte folgende Punkte:

- Bestellungen per Nachnahme oder Vorrauskasse
- Für jede Diskette ergibt sich ein Unkostenbeitrag von DM 8.-
- Pro Sendung kommt ein Versandkostenbetrag (für Porto und Verpackung) von DM 5.- (Ausland DM 10.-) hinzu.

*Bei einer Bestellung von 5 oder mehr Disketten entfällt der Versandkostenbeitrag!*

- Bei Nachnahme zuzüglich 3.70 DM

Anschrift:

MAXON Computer GmbH
KICKSTART PD
Postfach 55 69
6236 Eschborn

Die Diskettenbestellung kann auch telefonisch erfolgen. Der Versand erfolgt dann per Nachnahme.

Tel.: 06196 / 48 18 11
(Mo.-Fr. von 9⁰⁰-13⁰⁰ u. 14⁰⁰-17⁰⁰ Uhr)



# KICK 130-94

## KICK 130: ANWENDUNGEN

**SUPERLIGA**: Fußballergebnisverwaltung mit 577 Mannschaften aus 23 Ländern und 34 Ligen. Über 10000 Ergebnisse können gespeichert werden. Das Eingeben gestaltet sich sehr einfach. Aus den bisherigen Ergebnissen kann eine Prognose (Regressinsanalyse) für kommende Spiel und die Meisterschaft erstellt werden. Autor: Rolf Morlock

## KICK 129: SOUNDS

**Neue Kompositionen** von Andreas Starr aus Gronau, erstellt mit Sonix. Effektvolle Stereostücke mit Player.

## KICK 128: SOUNDS

**Scores, Drumkits und Soundfiles** für Dynamic Drums von New Wave Software.

## KICK 127: GRAFIK

**Mandelbrot-Show II**: dies ist der zweite Teil der Reise durch die Welt der Mandelbrotgrafiken. Autor: R. Beck

## KICK 126: GRAFIK

**Mandelbrot-Show I**: eine sehr informative Reise durch die Welt der Mandelbrotgrafiken. Jedes Bild wird ausführlich erläutert und das jeweils darauffolgende ist eine Vergrößerung eines Ausschnitts davon. Mit Mandelbrotgenerator MSE. Autor: R. Beck

## KICK 125: GRAFIK

**SLIDESHOW**: Komfortables Diashow-Programm für DPaint-Bilder (320x200). Alle Bilder des aktuellen Direktories werden nacheinander auf Mausklick angezeigt. **DARK**: Copper-Demo, die viele Elemente eines Ballerspiels enthält. C-Source. **GLOBE**: Eine kleine drehende Weltkugel. PopUp-Menu, Source in C. **ICONS**: verschiedenste, animierte Icons (Disk, Drawer, Tool, Project, Garbage). Sehr originell und lustig!

## KICK 124: ANWENDUNG

**STARCHART**: umfangreiches und interessantes Sternenprogramm mit unendlich vielen Einstellungen. Mit der Maus können Informationen über die Sterne abgerufen werden. **AIRFOIL**: ein einfacher Windkanalsimulator, verschiedene Einstellungen.

## KICK 123: UTILITIES

Neue Utilities, die aus den aktuellen Einsendungen zusammengestellt werden. Hier gibt es immer ein riesiges Angebot, denn der AMIGA läßt viele Möglichkeiten für Hilfsprogramme.

## KICK 122: MODULA (AMOK 12)

Auch auf dieser AMOK-Diskette gibt es wieder jede Menge Routinen und Programme: Modul zum Ansteuern des Fischertechnik-Interfaces, ein schnelles und sicherers HD-Backup, Disassembler für M2Amiga-Object-Files, ArraySort, Janus, Timer, StripText, ConvLibSym, MakeInLine, Diff, Tips zum M2-Compiler

## KICK 121: MODULA (AMOK 11)

**R.o.M.**: ein komfortables Mathematikprogramm und Funktionsplotter. **TrackDiskSupport**: erleichtert Programmierung der TrackDisk. **M2Druka**: Druckutility für ASCII-Dateien. **Beep**: erzeugt akustisches Warnsignal.

## KICK 120: UTILITIES

PCopy, HexCalc, Dux5 (CLI-Utility), SDBackup, FlamKey (Computerschloß), VirusX, SCT (Farbeinstellung von WB), Surveyor, HPMan (totale Kontrolle über den Laser)

## KICK 119: AMOK #10

FlightDemo u. 3D-Demo (Echtzeitdemonstrationen bewegter räumlicher Grafik - super), FileRequester, SuperLists1.3, Speech (dt. Sprachausgabe), Feigenbaum.

## KICK 118: AMOK #9

GraphicSupport (Makros für Graphics-Primitives/Intuition), PrintIt2.0 (Hardcopyroutine für Epson), M2Test (Dokumentation über Fehler bei ARRAYs mit Index BOOLEAN), TextFont (Laden v. Zeichensätzen), Sounds und Graphics.

## KICK 117: AMOK #8

MemSystem (Speicherverwaltung), PrinterSupport (unterstützt Printer.device), IDCMP (Hilfsroutinen), Blitter (direkte Blitterprogrammierung), SpaceDemo (3-D Demo), IFF8SVXLoad (Laden/Abspielen gesampelter Sounds).

## KICK 116: AMOK #7

Reminder (Wecker für Programmierer), WarpText (schnelle Textausgaberoutine), MuchMore (Softsroll), M2Test (Hinweise auf Fehler in den Bibliotheksmodulen), SoundTask, AVL-Bäume, Listen, Queues, Stack.

## KICK 115: SPIELE

**CHEES-BOXES** ist das allseits bekannte "Käsekästchen"-Spiel, das man jedoch nicht mit dem folgenden verwechseln sollte. 2-4 Spieler, mausgesteuert, AMIGABasic. Autor: J.D. Mallander. **VIER GEWINNT**: zwei Spieler, mausgesteuert, AMIGABasic. Autor: Björn Kessels. **COSMIC**: Strategiespiel. Source in C. Autor: Carl Edman. **SOLITAIRE** löst beliebige Stellungen des bekannten Spiels. Source in C. Autor: Thomas Lagally **GRID IT**: spannendes Rennen für zwei Spieler. Benötigt Joystick, Source in C. Autor: Jörg Tuttas **LABYRINTH II**: Textadventure in Stil der INFOCOM-Spiele. Englisch, C.

## KICK 114: BILDER-SHOW

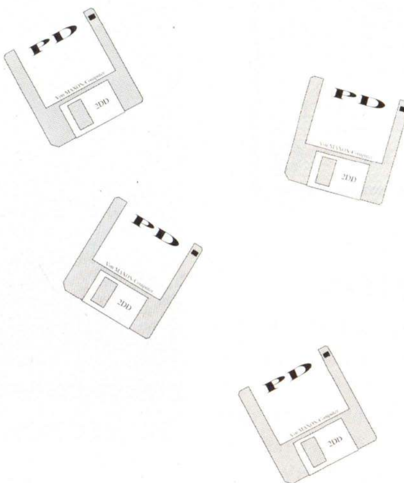
Eine Grafik-Show mit Mandelbrotbildern und Colorcycling. Dadurch wird die Wirkung der eindrucksvollen Bilder hervorgehoben. Autor: Hermann Dörries, Wildeshausen

## KICK 113: SOUNDS

Auf dieser Diskette ist der zweite Teil der KICKSTART SOUND DEMO III mit den Stereo-MIXES: Return to Defender, Dark Castle Mix, Chicago-Tune, Disco-Mix und Crazy Tune. Wichtig: Die Diskette läuft nur mit KICK 112!

## KICK 112: SOUNDS

Zum dritten Mal haben unsere Soundspezialisten TOB & TWP zugeschlagen. Das Ergebnis ist die KICKSTART SOUND DEMO III mit einer Menge hervorragender Sounds auf zwei Disketten (siehe KICK 113). Mit von der Partie sind in der Rubrik CHARTS: Relax, Jean Michelle Jarre, Every Breath You Take, Axel F., Push, Tell It To My Heart und Blue Monday. Alles HiFi-Stereo mit einer extralangen Spielzeit.




## KICK 111: BILDER

20 wunderschöne IFF-Grafiken von Frauenprofilen. Mit Diashow, sehenswert!

## KICK 110: UTILITIES

**MEMORYCLOCK:** zeigt Uhrzeit, Datum, RAM- und Chip-Memory. **DATAMAKER, IFF-CONVERTER**: wandelt Dump-Format-Bilder in IFF-Format. Source zu allen Programmen in Assembler, Autor: Roger Fischlin.
**SDBACKUP**: Festplatten-Backupprogramm, das die komplette Platte in komprimierter Form auf Disketten abspeichert. Viele Optionen. **RUNBACK:** läßt Programme im Hintergrund laufen, damit sie keine CLI-Fenster benötigen.

## KICK 109: SCHULE

**VOK-BOY V1.0:** Vokabeltrainer in AmigaBasic. Autor: Michael Hennemann
**WORD:** Fremdsprachenlernprogramm, das Begriffe und ganze Sätze abfragt. Mit umfangreichen Lektionen in Englisch und einigen in Französisch. Kann leicht erweitert werden. Verwaltet beliebige Sprachen, sehr komfortabel. Autor: Daniel Neukomm, Bern (Schweiz)

## KICK 108: BILDER

Viele schöne Bilder zum Anschauen und Weiterbearbeiten.

## KICK 107: ANIMATIONEN

Weitere Animationen von Tobias Richter: JET (Düsenjet-Landeanflug), RELIANT (ein Raumschiff im frontalen Anflug), NELSON (rotierendes Raumschiff). Die Animationen demonstrieren, wie eindrucksvoll die mit VideoScape erstellten Sequenzen sein können. Sehenswert!

## KICK 106: ANIMATIONEN

Tobias Richter hat uns wieder einige seiner mit VideoScape 3D-Animationen geschickt: LOTUS (der bekannte Lotus in einer neuen Sequenz), WILKERSON, DURETT und RELIANT (Raumschiffe in schneller Rotation).

## KICK 105: ANIMATIONEN

**LLEWELLYN** ist eine mit dem Sculpt 3D-Animator erstellte Filmsequenz mit musikalischer Untermalung. Eine schöne Demo dieses Programms von Michael Clinard.

## KICK 104: ANWENDUNGEN

**DIRECTORY MASTER V1.1:** Dieses Programm bringt Ordnung in Ihre Diskettensammlung. Die Dateien jeder eingelegten Diskette werden automatisch in eine Liste übernommen. Diese kann dann auf komfortable und vielfältige Weise bearbeitet und verändert werden. Sehr viele Funktionen, einfache Bedienung, sehr leistungsfähig und schnell. Als Zugabe gibt es die jeweils aktuelle Liste des KICKSTART PUBLIC DOMAIN SERVICE.
**MEMOPAD:** ist ein echt nützliches Hilfsprogramm für vergeßliche AMIGA-Anwender. Hier können wichtige Termine und Daten festgehalten werden. Bindet man den Befehl CHECKMEMO in die Startup-Sequence ein, dann erinnert das Programm an diese Termine. Einfache Mausbedienung, sehr komfortabel und hilfreich.

## KICK 103: GRAFIK

**C-LIGHT**: ein sehr einfach zu bedienendes, mausunterstütztes Raytracer-Programm. Die Einstellungen erfolgen über viele Regler und Schalter. **NEWJUGGLER**: ein Grafikprogramm, das den Eric Graham's JUGGLER innerhalb eines Bildes ablaufen läßt. Source in C; Autor: Peter Weiland, Wietze. **HAM'S**: einfaches Malprogramm für den HAM-Modus in AmigaBasic. Das Programm kann leicht geändert werden, da der Source-Code gut strukturiert ist. Autor: Henning Frommer, Vallendar

## KICK 102: SPRACHEN

**STONY BROOK PROLOG**: auf dieser Diskette befinden sich die Source-Codes (Lattice C 4.0 und Prolog) des PD-Prolog-Systems von Diskette 101.

## KICK 101: SPRACHEN

**STONY BROOK PROLOG** Version 2.3.2: Dieses PD-Prolog-System basiert auf einem System, das ursprünglich für UNIX-Rechner konzipiert wurde. Das System beinhaltet sowohl einen Interpreter als auch einen Compiler (bis zu 3000 LIPS). Der Sprachumfang entspricht weitgehend dem Edinburgh-Standard. Das System besitzt zusätzlich einige Besonderheiten z.B. dynamische Prädikateinbindung, Mischen von interpretiertem und compiliertem Kode, Makros und andere Hilfen. Inklusive ca. 100seitiger Dokumentation. Benötigt mindestens 1MB Speicher.

## KICK 100: JUBEL-DISK

Auf dieser Diskette sind Programme zusammengefaßt, von den wir denken, daß sie für einen AMIGA-Besitzer unverzichtbar sind. Die Palette reicht von Kopierprogrammen, CLI-Hilfen bis zu Spielen. Lassen Sie sich überraschen - es wird sich lohnen!

## KICK 99: SPIELE

**BrainWork**: Puzzle. *Autor: Timmy.* **Amiga-Wurm**: Tron-Variante; AmigaBasic. *Autor: Christoph.* **May Valley of the Aztec's**: ähnlich "Kaiser"; ABasic. *Autor: Matthias Hakuba* **Bauernskat**: Source in AmigaBasic. *Autor: Oliver Peter* **Breakout**: 99 Level mit Editor; Source in C. *Autor: Peter Händel.* **Kniffel**: bekanntes Würfelsspiel für 4 Personen oder gegen Computer; sehr schöne Grafik. Mausbedienung. *Autor: Michael Teistler.* **PACCIE**: Pacman-Variante in Basic.

## KICK 98: SUPER-SOUNDS

**KICKSTART SOUND DEMO II:** sicherlich eine der besten Sounddemos, die es derzeit auf Disk gibt. 18 programmierte Musikstücke, aktiongeladener Sound, Stereo, sehr lange Spielzeit, kurze Ladezeit durch gepacktes Format, Sound-Slideshow. *Komponist: Tom Beuke*

## KICK 97: UTILITY

**TASKMON V0.4**: zeigt die Zustände aller Tasks; Source in C. *Autor: P. Erpenbeck.* **FilterSwitch 2.0**: damit läßt sich der Tiefpaßfilter per Alt-Taste an- und ausschalten. **DMASwitch 2.0**: Bildschirmabschalter; Source in C. *Autor: Andreas Jung.* **MouseTool** (Fenster hervorholen, vergrößern, verkleinern, PALPatch (verändert Intui.-Lib., vergrößert alle Screens und Windows auf PAL-Größe), ChangeTaskPri, CLI4WB (CLI-Programme von der WB starten); alle Prg. mit Source (Profimat-Assembler). *Autor: Richard Englert, Fürth.* **RossiDress V1.0**: benutzerfreundliche Adressenverwaltung; AMIGABasic. *Autor: Jürgen Rogg.* **Milben**: kleiner Screenhack; Source in C. *Autor: Guido Appenzeller*

## KICK 96: UTILITY

**TurboBackup**: komfortables und schnelles Kopierprogramm; kopiert auf bis zu 3 Laufwerke mit Verify.
**MRBackup2.1**: Harddisk-Backupprogramm, sichert auch überlange Dateien.
**DMouse**: Bildschirmschoner, Mausabschalter, Mausbeschleuniger, Fensteraktivator, Funktionstastenbelegung.
**ARC**: Programm zum Packen und Entpacken von Dateien. Hiermit können Programme platzsparend abgelegt werden (siehe PD 94/95).

## KICK 95: ANWENDUNGEN

**AnalytiCALC**: professionelles Tabellenkalkulationsprogramm (Spreadsheet), das auch schon für MS-DOS, VAX und PDP-11 existiert. Das Programm arbeitet im Interlace-Modus und benötigt deshalb 1MB Speicher. Sehr leistungsfähig (18.000 * 18.000 Zellen usw.).
***Wichtig: Die Dateien der Disketten 94 & 95 sind in komprimierter Form abgelegt. Sie benötigen das Programm ARC (PD 96), um sie wieder zu entpacken.***

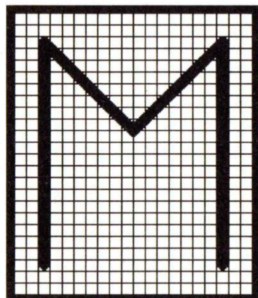
## KICK 94: ANWENDUNGEN

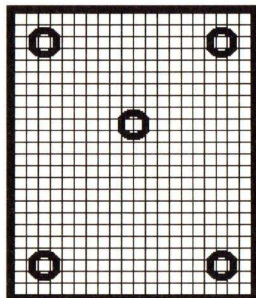
**RIM-5** ist ein relationales Datenbanksystem, für Verwaltung großer Datenmengen. Integrierte Programmiersprache. Help-Funktion. Incl. 80-seitiger Anleitung und Source in Absoft-Fortran .

# VORSCHAU

## Schrifterkennung

In einem ausführlichen Grundlagenbericht erfahren Sie, wie ein 'Auge' des Computers Schrift analysieren und weiterverarbeiten kann. Das Einsatzgebiet solcher Anwendungen ist schier unüberblickbar, doch Probleme bei diesem System sind zahlreich. Wie sie gelöst werden? Mehr im nächsten Heft.





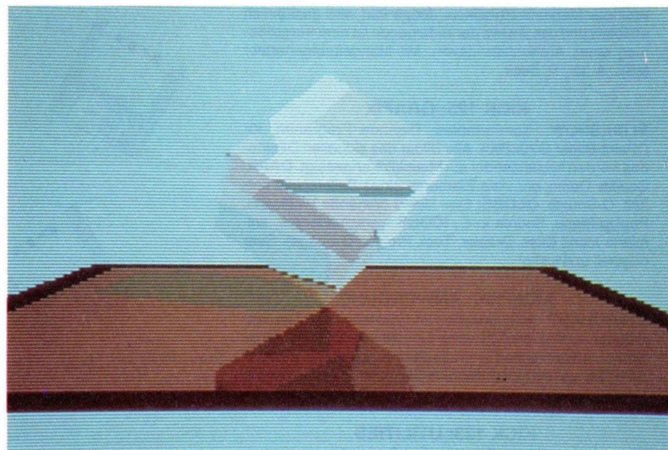
## DBW-RENDER und ANIMATION

Im nächsten Heft stellen wir das Raytracing-Programm DBW Render vor. Das Besondere an diesem Raytracer ist, daß er Public Domain, demzufolge für wenig Geld zu haben ist. Der Nachteil an diesem Programm war bisher die fehlende Animationsmöglichkeit, aber dies gehört mit dem nächsten Heft der Vergangenheit an. Wir stellen ein Programmpaket vor, das fließende Animationen aus Ihren DBW-Render-Bildern erzeugt.

*Ab 16. Juni an Ihrem Kiosk.*

## Optische Maus

Im Lieferumfang des AMIGA findet sich eine MAUS, die das Arbeiten mit dem Computer erleichtert und vereinfacht. Doch ist die mitgelieferte AMIGA-Maus nicht das Gelbe vom Ei. Da sie mechanisch stark beansprucht wird, treten besonders mit ihren Abnahmerollen Probleme auf. Solche Probleme kennt die optische Maus nicht, die keinerlei mechanischen Anforderungen ausgesetzt ist. Zusätzlich hat sie drei Mausknöpfe, die in späteren Betriebssystemversionen unterstützt werden sollen. Wie sich die Maus im praktischen Einsatz bewährt hat, lesen Sie in der nächsten Ausgabe.



## Und was lesen Sie sonst?

Wie immer können Sie in unserem ausführlichen Spieleteil alles neue erfahren, was dieses Genre betrifft. Interessante 'KICKS FÜR INSIDER' finden Sie ebenfalls. Natürlich haben wir wieder einiges in Erfahrung bringen können, was Sie sicherlich interessiert.


**Impressum**

**KICKSTART**

**Chefredakteur:**
Uwe Bärtels (Chefredakteur)(UB)
Markus Nerding (Stellvertreter) (MN)

**Redaktion:**
Andreas Krämer (AK)
Harald Schneider (HS)
Marcelo Merino (MM)
Harald Egel (HE)
Christian Keller (CHK)

**Herausgeber:**
MAXON-Computer GmbH
Industriestraße 26
6236 Eschborn
Tel.: 06196/481811
FAX: 06196/41137

**Redaktionelle Mitarbeiter:**
Gerald Carda (GC)
Jobst Hermeier (JH)
Carsten Borgmeier (CBO)
Martin Silbernagl (MS)
Sven Stillich (SV)
Florian Du Bois (FDB)

**Redaktionsanschrift:**
MAXON-Computer
KICKSTART
Postfach 5569
6236 Eschborn
Tel.: 06196/481811

**Public Relations:**
Claus Peter Lippert

**Auslandskorrespondenz:**
Derek Dela Fuente (ddf/GB)

**Titelbild:**
Gabriele Imhof

**Fotografie:**
Klaus Ohlenschläger Foto & Design
Bernd Opitz

**Illustrationen:**
Barbara Jacobs

**Comics:**
Pit Burkhard

**Produktion:**
Gabriele Imhof
Karl-Heinz Hoffmann

**Druck:**
Ferling Druck, Darmstadt

**Verlag:**
Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt 13
Tel.:06151/56057
FAX:06151/55689 - 56059

**Verlagsleitung:**
Hans-Jörg Heim

**Anzeigenverkauf:**
Kyriakulla Margaritis
Uwe Heim (Ltg.)

**Anzeigenpreise:**
nach Preisliste Nr.3, gültig ab 1.1.88

**Bezugsmöglichkeit:**
Zeitschriftenhandel, Kauf- und Warenhäuser, Commodore-Fachhändler oder direkt beim Verlag.
KICKSTART erscheint 11 mal im Jahr

Einzelpreis: DM 7,- ,ÖS 56,- SFr 7,-
JahresabonnementInland: DM 70,-
Europ. Ausland DM 90,-
Luftpost DM 120,-



## HD 3000

externes Festplattensystem für alle Amiga überragende Leistungmerkmale wie Datentransferrate bis **327K/sec** ● NEC Qualitätsplatten ● förmschönes Gehäuse als Monitorunterbau ● eigenes Netzteil ● Lüfter ● sehr komfortabeles Format-Menue ● defekte Zylinder werden erkannt, gesperrt und erstetzt, so erhalten Sie auch bei Harderrors immer die volle Plattenkapazität und brauchen nicht nach Fehlerrate Ihre Partitionen zu wählen

| | |
|---|---|
| 20 Megabyte | **998.-** |
| 30 Megabyte | **1098.-** |
| 40 Megabyte | **1498.-** |
| 60 Megabyte | **1698.-** |



## Golem Memory Station

externes Festplattensystem für alle Amiga
technisch mit den gleichen überragenden Leistungsmerkmalen wie HD 3000 ● zusätzlich kann eine weitere Festplatte oder ein 5,25 Zoll Laufwerk und ein 3,5 Zoll Laufwerk eingebaut werden ● Elektronik mit zwei Trackdisplays für die Laufwerke ist bereits vorhanden ● Im Harddisk Interface ist ein 2 Megabyte dynamik Ram Controller integriert
Nutzen Sie die Möglichkeit sehr preiswert Ram und Floppies nachzurüsten

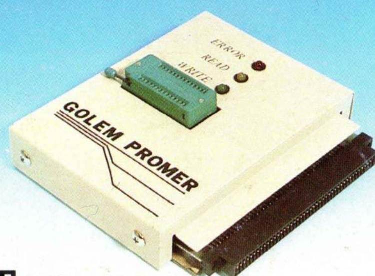
| | |
|---|---|
| 20MB | **1398.-** |
| 30MB | **1498.-** |
| 40MB | **1898.-** |
| 60MB | **2098.-** |

Nachrüstsätze zur Memory Station

| | |
|---|---|
| 2MB Rambank | **1098.-** |
| 3,5 Zoll Laufwerk NEC incl. Kabelsatz | **200.-** |
| 5,25 Zoll Laufwerk NEC incl. Kabelsatz | **220.-** |



**(02 31) 81 83 25-27**




## Eprommer

Eprombrenner für alle gängigen Eproms bis zum Megabit-Eprom leistungsfähiger integrierter Monitor ● komfortabeles Menue sehr schnell z.B. 27512 (64KB) in 15 Sekunden

Golem Eprommer incl. Brennsoftware u. Monitor **249.-**

**GOLEM**
HARDWARE

# AKTUELL

**8MB für A2000**
2MB bestückte Einsteckkarte ● 8MB gesokkelt ● bestückbar mit Megabit DRam autokonfigurierend **1298.-**

**Megabit DRam 51100 120nS**
für z.B. 8MB Karte **52.-**

**512K Karte für Amiga 500**
Einsteckkarte mit 512K Ram und Akkugepufferter Uhr **349.-**

**Viruskiller V2.0**
Das Superprogramm erkennt jeden gängigen Virus und beseitigt ihn, prüft Ihren Speicher auf Viren, schützt Ihre Disk vor erneutem Virusbefall, preiswerter Apdate Service
Einführungspreis nur **49.-**

**Hardware Virus Protector**
externes Steckmodul bietet absoluten Schutz vor jedem Virus, der sich unbemerkt auf Ihre Disk schreiben will **39.-**

**Netzteile 150W**
PC-Schaltnetzteil incl. Lüfter überall einsetzbar z.B. für Ihre Selbstbaufestplatte zum Sensationspreis **99.-**

Kupke Computertechnik · Burgweg 52a · D-4600 Dortmund 1 · ☎ 02 31/81 83 25-27

Entwurf + Druck PLUM, Düren-Oberzier, TEL 0 24 28/7 11